Modeling Musical Scores Languages

Louis Bigo

UNIVERSITÉ DE LILLE

École doctorale MADIS
Mathématiques, sciences du numérique et de leurs interactions

---

Modeling Musical Scores Languages

Modélisation du langage des partitions musicales

HABILITATION À DIRIGER DES RECHERCHES (HDR)

Spécialité : Informatique

présentée par

Louis Bigo

Centre de Recherche en Informatique Signal et Automatique (CRIStAL)
Équipe Algomus

soutenue publiquement le 27 mars 2023 devant le jury composé de

Frédéric Bimbot	Directeur de recherche CNRS, IRISA, Rennes	Rapporteur
Jean-Pierre Briot	Directeur de recherche CNRS, LIP6, Paris	Rapporteur
Gaël Richard	Professeur, Institut polytechnique de Paris	Rapporteur
Laetitia Jourdan	Professeure, Université de Lille, CRIStAL *(Présidente du jury)*	Examinatrice
Emilia Gómez	Professeure, Université Pompeu Fabra, Barcelone	Examinatrice
Mathieu Giraud	Directeur de recherche CNRS, CRIStAL	Garant

Remerciements

Je souhaite en premier lieu remercier les membres du jury, Frédéric Bimbot, Jean-Pierre Briot, Emilia Gomez, Laetitia Jourdan et Gaël Richard, pour leur intérêt et la richesse de leurs critiques, qui contribueront j'en suis sûr à l'orientation de ces recherches. Je remercie particulièrement les rapporteurs pour leur lecture attentive du manuscrit.

L'aboutissement de cet HdR est le fruit d'un ensemble d'interactions avec de nombreux proches, collègues et amis, dont le soutien fût essentiel. Travailler avec eux constitue une chance précieuse et un plaisir authentique. Je souhaite les remercier chaleureusement.

Merci tout d'abord à Mathieu Giraud, au côté duquel j'apprend continuellement sur les plans scientifiques, musicaux et humains. Je pense qu'il constitue une chance pour ses collègues et pour notre équipe Algomus. Merci à Florence Levé et Mikaela Keller avec lesquelles j'ai le plaisir de m'ouvrir à de nouvelles thématiques scientifiques, la texture musicale et le traitement du langage naturel, qui constituent des éléments majeurs du travail présenté dans ce document. J'apprécie notre travail ensemble et j'espère qu'il se prolongera autant que possible.

Merci à toute l'équipe Algomus, Emmanuel Leguy, Richard Groult, Ken Deguernel avec qui travailler est un plaisir quotidien. Merci aux doctorants que j'ai, ou ai eu, la chance de co-encadrer, Laurent Feisthauer, Louis Couturier, Alexandre D'Hooge et Viet-Toan Le. Travailler avec chacun d'entre vous constitue une aventure riche et unique.

J'ai la chance d'être encouragé continuellement par ma famille et mes amis et je les en remercie chaleureusement.

Enfin merci Fanny pour ton soutien pratique, moral et intellectuel dans la préparation de ce manuscrit, qui met en perspective le langage et les mots, que tu connais si bien, et cette musique qui anime notre vie.

Parcours en recherche

- **Entre 2010 et 2013**, j'ai effectué à la suite du Master ATIAM une thèse en informatique à l'Université Paris-Est, au Laboratoire d'Algorithmique Complexité Logique (LACL) et à l'Institut de Recherche et Coordination Acoustique/Musique (IRCAM) sur le sujet *Représentations Symboliques Musicales et Calcul Spatial*. La thèse était dirigée par Olivier Michel, et co-encadrée par Antoine Spicher et Moreno Andreatta.

- **Entre 2013 et 2014**, j'ai été Attaché Temporaire d'Enseignement et de Recherche (ATER) à l'Université de Paris-Est.

- **Entre 2014 et 2016**, j'ai effectué un postdoctorat à l'Université du Pays-Basque à San Sebastian en Espagne, au sein du Music Informatics Group animé par Darrell Conklin. Mes recherches de postdoctorat ont porté sur l'élaboration de techniques de transformation de pièces musicales à l'aide d'algorithmes d'apprentissage automatique, dans le cadre du projet européen *Learning To Create*.

- **Depuis 2016**, je suis maître de conférences à la Faculté des Sciences et Technologies de l'Université de Lille. J'effectue mes recherches au sein de l'équipe Algomus du laboratoire CRIStAL dans le domaine de l'informatique musicale. Ces recherches portent sur l'élaboration de méthodes informatiques destinées à assister l'analyse et la composition de musique. Les méthodes utilisées se basent sur des algorithmes issus de l'apprentissage automatique ou de systèmes de règles, et portent sur des représentations musicales dites *symboliques*, c'est à dire rendant compte du contenu de la partition. La conception de ces méthodes implique généralement l'élaboration de représentations dédiées traduisant l'information musicale à différents niveaux d'abstraction allant de la simple note de musique jusqu'à la structure ou la fonction musicale. Cette thématique de recherche s'inscrit dans celles de la communauté internationale MIR (*Music Information Retrieval*).

Encadrements

Encadrement doctoral

- Entre 2017 et 2021, j'ai été co-encadrant de la thèse de **Laurent Feisthauer** (co-direction par M. Giraud, CNRS CRIStAL, et F. Levé, MIS UPJV) qui a effectué sa thèse dans l'équipe Algomus sur le sujet *Annotation automatisée des métadonnées structurelles dans les partitions musicales. Cas des modulations et des cadences pour la forme sonate*. La soutenance a eu lieu le 18 mai 2021. Cette thèse a donnée lieu à trois publications dans des conférences internationales (ISMIR 2018, ISMIR 2019 et SMC 2020), une communication nationale avec sélection sur résumé (colloque *Les Sciences de la Musique*, 2019), une communication internationale avec sélection sur résumé (*Digital Music Research Network*, 2019). Laurent a par ailleurs participé à la recherche et l'écriture d'un article d'équipe paru dans la revue TISMIR en 2019.

- Depuis octobre 2021, je co-encadre la thèse de **Louis Couturier** sur le sujet *Modélisation de la texture musicale dans les partitions pour piano* (soutenance prévue en 2024) avec F. Levé, directrice de thèse. La thèse de Louis, qui s'effectue dans le prolongement de son stage de fin d'étude, a pour l'instant donné lieu à une communication internationale avec sélection sur résumé (*Digital Music Research Network*, 2021) et deux articles dans des conférences internationales (SMC 2022 et ISMIR 2022).

- Depuis septembre 2022, je co-encadre la thèse d'**Alexandre D'Hooge** avec K. Deguernel (CNRS CRIStAL) et M. Giraud (directeur de thèse) sur le sujet *Élaboration d'outils d'intelligence artificielle pour assister la composition de tablatures pour guitare*. Les travaux de thèse d'Alexandre constituent le coeur du projet ANR JCJC TABASCO (2022-2026) dont je suis le coordinateur. Le sujet de la thèse d'Alexandre porte essentiellement sur l'élaboration d'algorithmes d'apprentissage automatique pour assister des situations ponctuelles dans le processus de composition, notamment la continuation de parties d'accompagnement par imitation de texture et l'ajustement automatique de l'expressivité résultante de l'emploi de techniques de jeux spécifiques à la guitare.

- Depuis octobre 2022, je co-encadre avec M. Keller et M. Tommasi (INRIA, CRIStAL, Université de Lille), la thèse de **Dinh-Viet-Toan Le** sur le sujet *Approches de Traitement Automatique du Langage Naturel dans le domaine musical : adaptabilité, performance et limites*. Les travaux de thèse de Dinh-Viet-Toan s'inscrivent dans le cadre d'une collaboration que j'ai engagé avec M. Keller en 2019, visant à mener une étude critique de l'utilisation d'outils issus du Traitement Automatique du Langage Naturel (TALN) dans le domaine musical, et à proposer des évolutions de techniques et de positionnement concernant cette pratique.

Encadrement de stages niveau Master

- Mathieu Kermarec, niveau M2, (co-encadré avec M. Keller), 6 mois en 2022
  Tokenisation de représentations symboliques musicales.
- Gabriel Loiseau, niveau M1, (co-encadré avec M. Keller), 4 mois en 2021
  What Musical Knowledge Does Self-Attention Learn?.
- Louis Couturier, niveau M2, (co-encadré avec F. Levé), 6 mois en 2021
  Modélisation de la texture pianistique. Analyse d'éléments texturaux symboliques dans les Sonates pour piano de Mozart.
- Quentin Normand, niveau M2, 6 mois en 2021
  Modélisation de techniques de jeu dans les tablatures de guitare.
- David Régnier, niveau M2, 6 mois en 2020
  Détection de la fonction musicale de la guitare dans les tablatures.
- Jules Cournut, niveau M2, (co-encadré avec M. Giraud), 6 mois en 2019
  Créativité musicale assistée par ordinateur : modèles pour l'aide à la composition dans Guitar Pro.

Résumé du manuscrit

Ce manuscrit décrit mes principaux axes de recherche autour du thème général de la *modélisation du langage des partitions musicales*. Ces travaux se situent dans la discipline de la Recherche d'Information Musicale (*Music Information Retrieval, MIR)*. Le manuscrit est organisé autour de trois chapitres, portant respectivement sur le répertoire classique, les tablatures de guitare dans le répertoire moderne populaire et l'application de techniques de Traitement Automatique du Langage Naturel dans le domaine musical.

Assister l'analyse musicale : modélisation du langage du répertoire classique

Le Chapitre 2 se focalise sur l'analyse musicale assistée par ordinateur dans le répertoire de la musique classique. Il rend compte entre autres d'une partie des travaux de la thèse de L. Feisthauer (2017 - 2021), sur la modélisation de la structure musicale, que j'ai co-encadré avec F. Levé et M. Giraud, ainsi que du travail de thèse de L. Couturier (2021 - 2024), sur la texture musicale, que je co-encadre actuellement avec F. Levé.

La première contribution propose une méthode élaborée dans le cadre de la thèse de L. Feisthauer pour la modélisation de cadences dans les partitions (Bigo et al., 2018). Les cadences correspondent à des phénomènes perceptifs conclusifs largement utilisés dans le style classique pour marquer la fin d'une phrase musicale. Ces éléments jouent un rôle essentiel dans la structure des pièces musicales et de manière plus générale dans les mécanismes d'anticipation que nous percevons à l'écoute de ces pièces. Un ensemble de 44 descripteurs issus de la théorie musicale sont utilisés pour calculer une représentation abstraite de la partition à chaque pulsation. Un classifieur est ensuite entraîné à corréler les valeurs de ces descripteurs avec un ensemble d'annotations expertes, permettant l'identification automatique de cadences dans de nouvelles partitions.

Nous décrivons ensuite un axe de recherche dédié à la modélisation de la *texture symbolique* dans les partitions pour piano. Les travaux de stage et de première année de thèse de L. Couturier ont permis l'élaboration d'une syntaxe décrivant la texture pianistique dans le répertoire classique à un haut niveau de précision (Couturier, Bigo, and Levé, 2022b). Cette description a lieu à l'échelle de la mesure. Elle rend notamment compte de la fonction musicale (mélodique, harmonique ou statique), des relations entre voix et de la présence d'un ensemble de figures musicales caractéristiques de ce répertoire. Cette syntaxe a déjà été utilisée pour la constitution d'un ensemble d'annotations ouvrant des perspectives pour l'entraînement de modèles d'apprentissage automatique dédiés à l'analyse de texture, la génération guidée par la texture et le transfert de style (Couturier, Bigo, and Levé, 2022a).

Ce chapitre décrit ensuite un algorithme pour la segmentation structurelle de partitions suivant le shéma général de la *Forme Sonate* (Bigo et al., 2017; Allegraud et al., 2019). De manière analogue à la détection de cadences, un ensemble de descripteurs dédiés sont implémentés et utilisés comme observations d'un modèle de Markov caché entraîné à identifier les bordures des différentes sections de la forme Sonate. On se penchera enfin sur le problème précis de la détection de *Medial Caesura* qui constitue un marqueur structurel essentiel de cette forme musicale (Feisthauer, Bigo, and Giraud, 2019).

Perspectives

Mes perspectives de recherche sur cet axe se concentrent essentiellement sur la modélisation de la texture pour piano qui fait l'objet du travail de thèse de L. Couturier. Elles comprennent en premier lieu la proposition d'une variété de métriques permettant d'évaluer la distance séparant deux textures symboliques. Il est notamment prévu d'expérimenter le potentiel d'algorithmes d'apprentissage non supervisé, tels que les auto-encodeurs variationnels, pour la construction d'espaces texturaux dont l'organisation interne aura vocation à refléter de manière intuitive l'appréhension de la texture par le musicien. Nous espérons par la suite exploiter ce type de représentations dans un cadre d'aide à la composition. Les liens fondamentaux entre texture et style musical permettront aux modèles élaborés dans le cadre de la thèse de L. Couturier d'être utilisés pour des tâches de transfert de style, par exemple dans le cas où la mélodie et l'harmonie d'une première pièce musicale sont conservés tout en adoptant la texture d'une seconde. La construction de modèles reflétant notre perception de la texture musicale pourra être orientée par la réalisation de tests d'écoutes lors desquels il est demandé aux sujets de quantifier la similarité ressentie entre des stimulis de texture variable.

Assister la composition musicale : modélisation du langage des tablatures de guitare

Le Chapitre 3 porte sur la composition musicale assistée par ordinateur, dans le cas particulier de la musique pour guitare dans le répertoire *moderne populaire*. Cet axe de recherche fait l'objet d'une collaboration industrielle avec l'entreprise Arobas Music qui édite le logiciel Guitar Pro et maintient le corpus de tablatures MySongBook auquel l'entreprise fournit à l'équipe Algomus un accès exclusif pour ces recherches, ainsi qu'avec le musicologue B. Navarret (Sorbonne Université, Iremus), spécialiste en pratique de la guitare. Ce chapitre décrit notamment les travaux de stage de J. Cournut et de D. Régnier, ainsi que la thèse d'A. D'Hooge (2022 - 2025) que je co-encadre actuellement avec K. Deguernel et M. Giraud. Cet axe de recherche fait enfin l'objet du projet ANR JCJC TABASCO (2022 - 2026) dont je suis le coordinateur.

La première contribution décrite dans ce chapitre consiste en un ensemble d'outils destinés à faciliter l'exploitation des données de MySongBook dans des tâches d'apprentissage automatique. Ces outils comprennent un parseur Music21 facilitant la manipulation de fichiers Guitar Pro dans le langage Python, ainsi qu'un ensemble d'encodeurs permettant la représentation des tablatures de guitare sous la forme de vecteurs binaires nécessaires pour l'entraînement de réseaux de neurones (Cournut et al., 2020). En prélude aux tâches d'apprentissage automatique qui constituent l'objectif central de ce projet, ces outils ont permis d'effectuer une étude statistique dans MySongBook rendant compte de propriétés variées sur les positions majoritairement utilisés par les guitaristes (Cournut et al., 2021).

Ce chapitre décrit ensuite une méthode destinée à l'identification automatique de la *fonction musicale* d'une tablature, en particulier sa vocation à jouer un rôle de premier plan, comme c'est le cas dans un solo, ou un rôle de second plan, comme c'est le cas dans une partie d'accompagnement (Régnier, Martin, and Bigo, 2021). Au delà de l'interprétation des résultats obtenus sur le plan musicologique, l'élaboration de cette méthode est motivée par la perspective de distinguer au sein de MySongBook

des sous-corpus consistant, facilitant l'entraînement de modèles destinés à assister la composition de tablatures associées à une fonction musicale précise.

La dernière contribution présentée dans ce chapitre présente une méthode pour la continuation de guitare rythmique par imitation de texture. Dans cette tâche, un modèle est entraîné à généraliser un style de texture, spécifié par un extrait de tablature de référence, à une séquence d'accords arbitraire. Se distinguant de la majorité des algorithmes présentés dans ce travail, cette méthode met en jeu un modèle destiné à apprendre la texture musicale de manière implicite à travers la description bas niveau des tablatures sur lesquelles il est entraîné.

Perspectives

Les perspectives de cette recherche constituent les axes du projet ANR TABASCO, dédié à l'élaboration d'outils issus de l'intelligence artificielle pour assister la composition de musique pour guitare, que je coordonne depuis le 1er octobre 2022 jusqu'au mois de septembre 2026. Le projet TABASCO comprend en premier lieu une étude sur les pratiques de composition de musique pour guitare dans le style populaire moderne. Cette étude sera menée en collaboration rapprochée avec notre collège musicologue B. Navarret. Elle aura pour but d'améliorer notre compréhension du rôle du logiciel de notation au sein du processus de composition dans les musiques de ce répertoire, ainsi que d'identifier les fonctionnalités logiciel qui pourraient contribuer au renouvellement de ces pratiques de composition. La suite du projet se concentrera sur l'élaboration et l'évaluation de ces fonctionnalités, principalement dans le cadre de la thèse d'A. D'Hooge. Guidée par les résultats de cette étude préliminaire, la partie algorithmique du projet se focalisera sur la composition de parties de guitare rythmique, en particulier sur la construction d'espaces de textures analogues à ceux pressentis pour la modélisation de la texture pour piano (thèse de L. Couturier). Poursuivant l'élaboration d'outils dédiés à la composition de parties d'accompagnement, le projet comprendra une partie se concentrant sur la génération de parties de guitare basse, respectant une séquence d'accords arbitraire, et conditionnées par un style spécifié par une pièce de référence. Le projet comprendra enfin un axe sur la modélisation des techniques de jeux, ayant pour but d'offrir au compositeur un contrôle fin sur l'expressivité de sa tablature. Le projet comprend enfin un ensemble de contributions open-source visant à faciliter la manipulation de tablatures de guitare par la communauté MIR, ainsi qu'une évaluation par des compositeurs des outils conçus au terme du projet.

Méthodes de Traitement Automatique du Langage Naturel pour la modélisation de partitions

Le Chapitre 4 présente un ensemble de réflexions et d'expériences visant à évaluer les apports du domaine du Traitement Automatique du Langage Naturel (TALN, ou *NLP* pour *Natural Language Processing*) pour la modélisation de partitions musicales. Ce sujet est motivé par le constat d'une utilisation croissante en recherche musicale d'algorithmes concus et pensés à l'origine pour le texte, contribuant à nous questionner sur la pertinence, le potentiel et les limites du rapprochement de ces deux

domaines. Ce projet de recherche inter-disciplinaire est co-supervisé par l'équipe Algomus pour l'informatique musicale, et l'équipe MAGNET pour le TALN, en particulier M. Keller avec qui j'ai initié ce projet en 2019. Il fait l'objet d'une collaboration bilatérale entre Algomus et l'équipe AMAAI de Singapore University of Technology and Design (SUTD) financé par un partenariat Hubert Curien de Campus France dont je suis coordinateur, et qui m'a permis de réaliser une visite de recherche de 10 jours au AMAAI en août 2022.

La première contribution décrite dans ce chapitre consiste en l'étude de la manière dont s'applique le principe d'*attention mutuelle* (*self-attention*) dans les partitions musicales (Keller et al., 2021; Keller, Loiseau, and Bigo, 2021). Dans le domaine du texte, pour lequel il à été définit à l'origine, ce mécanisme permet d'identifier des liens entre des mots distants, rendant compte par exemple de règles grammaticales subtiles. Bien qu'utilisé de manière croissante dans le domaine musical, l'interprétation du principe d'attention dans l'espace d'une partition fait encore l'objet d'une compréhension limitée. Le travail de stage de G. Loiseau propose un cadre permettant d'évaluer l'expressivité musicale de l'information stockée dans les valeurs d'attention d'un réseau de neurones *transformer*. La méthode proposée consiste dans un premier temps à entrainer un transformer sur une large base de données musicale, puis à utiliser le modèle entrainé pour extraire de manière systématique des relations d'attention associée à une nouvelle séquence musicale. L'expressivité de ces relations est ensuite évaluée à travers deux tâches musicales : la classification par compositeur et la détection de cadences.

La seconde contribution porte sur le principe de *tokenization* qui consiste à représenter la partition musicale sous la forme d'une séquence d'éléments atomiques, des *tokens*, afin de coller au cadre des séquences de mots du texte et permettre l'application directe d'algorithmes de TALN sur des corpus de partitions. Le travail de stage de M. Kermarec a permi d'élaborer et évaluer une méthode de tokenization invariante par transposition, dispensant le recours à l'*augmentation de données* par transposition généralement nécéssaire au modèle pour généraliser les connaissances musicales apprises de manière uniforme dans les différentes tonalités (Kermarec, Bigo, and Keller, 2022).

Perspectives

Les perspectives de ce projet de recherche correspondent essentiellement aux axes pressentis pour la thèse de D.-V.-T. Le (2022 - 2025). L'interdisciplinarité caractérisant ce projet nous encourage en premier lieu à dresser un état de l'art croisé réunissant des techniques de TALN, des représentations et tâches en MIR, ainsi que des expériences ayant croisé les deux domaines, dans le but d'identifier une cartographie des usages dans cette discipline. Dans la lignée du travail de stage de M. Kermarec, nous questionnerons et mènerons des expériences concernant la représentation séquentielle de contenu musical. Au delà de la sélection d'un dictionnaire de tokens appropriés à l'information musicale, nous nous focaliserons sur l'utilisation en musique d'algorithmes récents, tels que *WordPiece* et *Byte Pair Encoding* visant à ajouter au dictionnaire des *super tokens* représentant les n-grams prédominants dans un corpus de référence. Cette méthode a vocation à faciliter l'apprentissage par le modèle de connaissances abstraites, au prix d'une dépendance croissante au corpus d'apprentissage. Nous expérimenterons ces limites pour la modélisation d'un

style musical définit par un corpus d'œuvres de référence. Une partie importante de ce projet portera par ailleurs sur le potentiel du *transfer learning* en musique, où un modèle est pré-entraîné sur un large ensemble de données non étiquetées, puis ajusté pour une tâche aval spécifique pour laquelle n'est généralement disponible qu'un ensemble limité de données étiquetées. Enfin, ces travaux auront vocation à contribuer à une réflexion épistémologique générale sur les parallèles et divergences entre musique et langage naturel.

Contents

Chapter 1

Introduction

This manuscript presents research in the field of Music Information Retrieval that I have been working on over the past 7 years in the Algomus team of the CRIStAL laboratory at the University of Lille. The interdisciplinary field of Music Information Retrieval (MIR) aims at elaborating computational tools dedicated to the processing of musical data. MIR involves a variety of applications including among others computer assisted music analysis and composition. While MIR techniques are intended to process a variety of digital formats such as audio waveforms, the present research exclusively focus on the processing of symbolic representations of music, which are generally comparable to the content of a musical score.

The three axes of this manuscript are unified under the general notion of *musical language modeling*. Chapter 2 presents a variety of computational methods dedicated to the study of essential components the classical style. Chapter 3 focus on representations and algorithms aiming at modeling language elements of guitar tablatures in modern popular music. Chapter 4 presents experiments aiming at gaining some perspective regarding the application of Natural Language Processing (NLP) methods to address musical tasks.

This introduction presents selected aspects of MIR, which play an important role in the experiments presented in the next chapters. After a brief overview of representations, datasets and tasks, we will present a common machine learning approach that has been used for most of the research presented here.

1.1 Processing musical data

1.1.1 Computer music representations

As illustrated in Figure 1.1, musical data can be collected into digital corpora mostly in three possible types of representations: audio, images and symbolic. Accessing music information in audio signals and score images require specific signal processing (Muller et al., 2011) and optical recognition tools (Calvo-Zaragoza, Jr, and Pacha, 2020). In particular, the processing of musical audio data commonly involves the Fourier transform which converts a signal that depends on time into a representation that depends on frequency. Frequency information are commonly processed with dedicated representations including spectrograms and chroma features, which facilitate the analysis of some higher-level musical aspects including timbre and pitch. The use of Fourier analysis in music processing is largely described for research and pedagogy purpose by Müller (2015). Signal processing tools dedicated

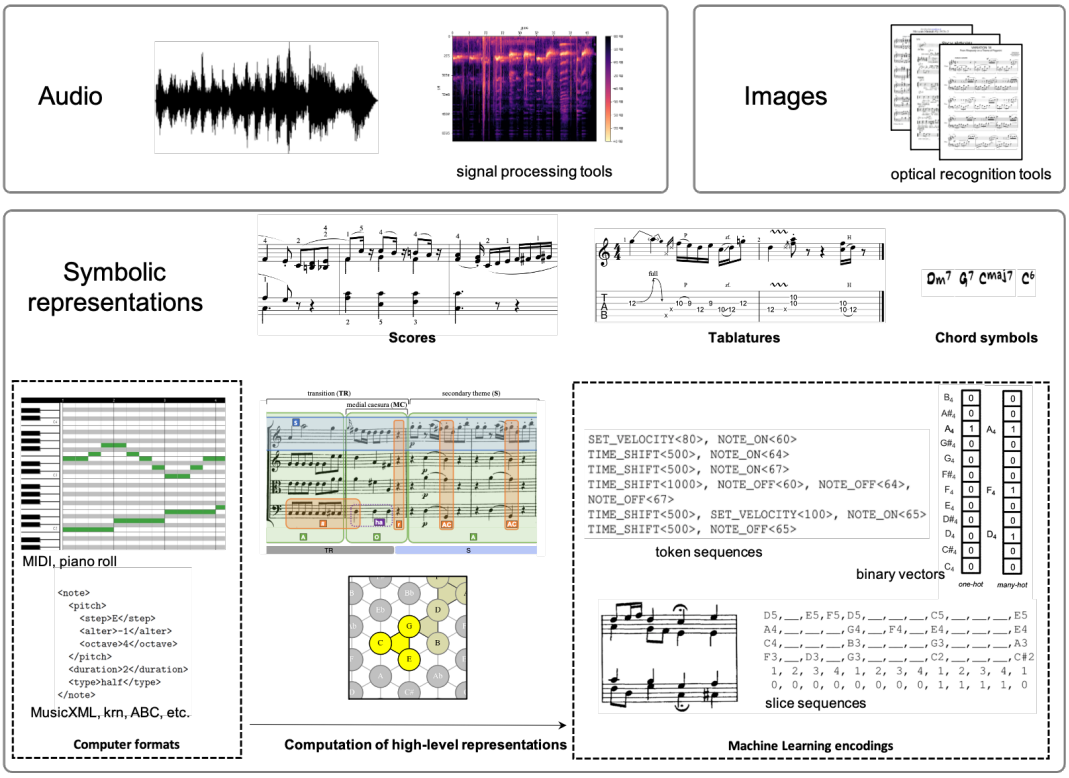


FIGURE 1.1: Different types of musical representations. Computer formats of symbolic representations can be processed to extract higher level musical information. Different levels of music representations can be encoded as simple data structures such as vectors and sequences, to be processed by ML algorithms. The figure includes figures from (Müller, 2015; Hadjeres, Pachet, and Nielsen, 2017; Huang et al., 2019; Briot, Hadjeres, and Pachet, 2019)

to MIR research are today provided through programming language libraries such as the Librosa (McFee et al., 2015) and Essentia (Bogdanov et al., 2013) packages.

In contrast, symbolic representations explicitly encode music information at a higher level, for instance the level of notes and chords. Symbolic formats include text formats such as MusicXML, MEI, ABC, and Humdrum **kern, as well as the binary MIDI format. A variety of tools have been elaborated to facilitate the computational processing of symbolic music representations. The music21 python package (Cuthbert and Ariza, 2010), which has been used in most of the works presented in the next chapters, encodes score data as complex data structures with a high level of precision. In contrast, the Partitura package (Cancino-Chacón et al., 2022) allows lightweight extractions of selected musical features intended to be processed by ML (Machine Learning) algorithms.

The application of common ML algorithms, such as neural networks, requires data to be represented as simple data structures such as sequences and vectors of binary values. Briot, Hadjeres, and Pachet (2019) review most approaches that are used in MIR to represent musical data as sequences of binary vectors compatible with the training of machine learning models. Generative models, such as the texture imitation model presented in Section 3.3.5, generally require to be trained on vectors representing low level information in order to have the ability to output

complete musical information in the sense that the result includes enough information to encode a content that can be listened. In contrast, music analysis models as those presented in Chapters 2 and 3, can process vectors of higher level representations, possibly with a higher loss of information, as they are not expected to produce listenable content on their output.

Although the research described in this manuscript is limited to the processing of symbolic representations, recent progress in audio-to-score transcription (Hawthorne et al., 2019) and optical music recognition (Calvo-Zaragoza, Jr, and Pacha, 2020) seem to suggest possible future benefits of symbolic music algorithms on much larger corpora, originating from large audio and score image file collections. Alternatively, deep-learning techniques seem to open the door to new approaches mixing both types of representations. This is for example the case with transfer learning where a model can be pretrained with symbolic data to improve its accuracy for some audio-to-score transcription task (Liang, Fazekas, and Sandler, 2019). In a general way, it seems that the popularization of neural networks in all fields also seems to contribute to bridge the gap between these MIR communities which used to employ much different tools, for instance signal processing methods for audio data and music theory-driven expert rule systems for symbolic data.

1.1.2 MIR datasets

Computational music analysis and generation tasks commonly target the modeling of a musical style. The availability of representative stylistic datasets is crucial for the evaluation of style models as they enable to measure their accuracy by confronting them to real musical data. In case of machine learning approaches, datasets are also required to train and parameter models. For these reasons, the MIR community has raised a number of research initiatives dedicated to the transcription and annotation of curated corpora[1] intended to support computational experiments in music analysis and generation.

A reasonable stylistic uniformity is commonly assumed for a corpus of works in the same instrumental configuration (*e.g.*, string quartets or piano solo) composed by one single composer, although this uniformity naturally tends to vary across the pieces forming the corpus. Corpora in a given instrumental configuration include for instance four-voice chorales, piano sonatas or string quartets, from famous composers such as J.-S. Bach, J. Haydn, W.A. Mozart and L.v. Beethoven. Beyond western classical music, the MIR community also puts substantial efforts in the creation of corpora of modern music, including popular and jazz, as well as under-represented cultural musical styles. Table 1.1 lists the different corpora that have been used in the research described in this report.

In addition to the raw encoding of musical content, such as audio waveforms, score images and textual score descriptions, MIR research makes use of expert annotations which label musical content at various time granularity, to indicate high-level musical information, for instance relating to key, structure or style. Expert musical annotations are however often expensive to produce as they generally require specific musical skills and substantial time resources. To facilitate this process, a number

[1] https://www.ismir.net/resources/datasets/

of annotations tools are elaborated in MIR, such as the Dezrann web platform[2] developed by the Algomus team (Garczynski et al., 2022).

Datasets with expert annotations may thus be used as a reference to train and evaluate MIR models, even if these annotations are often subject to debate between experts given the inherent ambiguity and subjectivity of music. Experts may also have different analytical methods, resulting in distinct datasets. Ideally, MIR tools and models should be able to take into account this variety in the expert views.

1.2 Computer-assisted music analysis and composition

Music analysis and composition are related, as they may use the same concepts and features. It is known that musicians may benefit from this relation by learning and practicing both tasks. Analogously, analysis and generative algorithms are likely to be based on common representations, and models. In the next two sections, we briefly discuss some aspects of these fields, highlighting specific MIR challenges.

1.2.1 Computer-assisted music analysis

Computational Music Analysis gathers research aiming at collecting data and elaborating representations and algorithms to assist or automate music analysis. This section details several benefits of such approaches in music analysis.

First, computational approaches allow the processing of large volumes of data that would be impossible to process manually. This facilitates for instance the exhaustive study of a musical repertoire and the drawing of representative conclusions. Secondly, the ubiquitous use of computer science in a variety of fields suggests the adaptation of algorithms that have been conceived to process other data in other domains such as image, text or biology, providing original approaches to music analysis. Third, data-driven experiments incidentally allow the confirmation or refutation of some musical intuitions. For instance, we addressed cadence detection in Section 2.3.1 by implementing expert knowledge features mentioned in music theory writings and that we felt correlated with cadence occurrences. This process allowed a fine study of each of these features and highlighted unexpected correlations and disconnections with cadence occurrences.

Although computational music analysis research frequently targets the implementation of predictive algorithms, these tools rarely seem directly used by musicologists thereafter. There are more chances however for these algorithms to be used, or to inspire, building blocks of more general systems dedicated to music analysis or even generation. For instance, a model intended to generate music could benefit from a specific step dedicated to the building of sophisticated cadences at the end of musical phrases. Finally, and from a more general musical point of view, the simple evaluation of an algorithm intended to model a musical concept provides presumable insights regarding the complexity of this concept. The evaluation of our cadence detection model in Section 2 showed for instance that half-cadences appear to be more complex objects than authentic cadences.

[2] http://www.dezrann.net

1.2.2 Computer-assisted music composition

Elaborating computational tools to assist or imitate music composition has been the purpose of multiple research since the beginning of computer science. This includes for instance the Illiac Suite by Hiller and Isaacson (1959), which is often considered as the first computer generated piece. Music composition has motivated the conception of a variety of generative algorithms elaborated in the MIR community, which have been comparatively described in different surveys including those of Herremans, Chuan, and Chew (2017) and Briot, Hadjeres, and Pachet (2019). Aiming at facilitating the use of algorithmic tools by composers, a number of composition dedicated software has been developed including the visual programming language OpenMusic (Bresson, Agon, and Assayag, 2011) and the MAX library bach (Agostini and Ghisi, 2015). Alternatively, integrating composition algorithms as functionalities of composition environments has become a major concern in the last years with several attempts within music production software such as Ableton Live (Esling et al., 2019; Roberts et al., 2019) or within music notation software such as MuseScore (Hadjeres, Pachet, and Nielsen, 2017) or Finale (Lupker, 2021). From a broader perspective, the use of computational tools by composer is becoming itself an active topic of discussion and research (Kayacik et al., 2019; Ben-Tal, Harris, and Sturm, 2021; Deruty et al., 2022).

Compositional practices however considerably vary among composers, including within a same musical style. Designing tools to assist music composition therefore faces in the first instance the challenge of identifying common composition situations for which such tools might be beneficial. One way to address this challenge is to limit the tool's role to punctual steps of the composition process. The designed tool will then be limited to one particular musical task, such as melody creation, harmonization, orchestration, voicing, texture designing, structure building, or notation. The texture imitation method presented in Section 3.3.5 follows this idea, by proposing to the composer the rendering of a given chord symbol in the style of a reference texture.

In contrast with these task-specialized tools, another part of MIR research attempts to build autonomous generative systems intended to produce musical pieces in the same manner composers do. The recent progress in artificial intelligence in the last decades have particularly contributed to promote this practice[3], reaching to an increasing proportion of MIR research focusing on this task. My research interests in this field however are more in line with the first approach, in which algorithms are intended to be used as punctual tools by the composer instead of aiming at substituting him.

1.3 A common ML prediction framework

An important part of the research described in this document consists in prediction tasks that have been addressed with a common machine learning methodology which is described in this section. Given a corpus of musical scores, on which a particular kind of labels have been manually annotated by expert musicologists, a

[3] As presented by Briot, Hadjeres, and Pachet (2019), music generation has been the purpose of experiments with major deep learning techniques elaborated in other applications including CNNs, RNNs, transformers, GANs or VAEs.

light[4] machine-learning classifier is set up and trained to predict occurrences of these manual labels on the corpus scores, which are represented in an abstract way. These abstract representations result from the computation of high-level musical features judiciously selected according to the targeted task. Examples of such features are *mean pitch*, or *presence of a major triad*. Selecting features is commonly performed jointly with a reflection on the *time granularity* at which these features must be computed. While signal processing tasks will commonly compute features in an absolute time granularity (*e.g.*, every millisecond or every second), symbolic MIR will generally use the time of the score, computing features for instance at each beat or each bar of the musical score.

The training of light machine learning models on high-level representations has somehow become a specificity of the Algomus team over the past years, including applications in cadence, structure or texture prediction. This approach contrasts however with a predominant tendency in MIR research, which consists in training deep models on low-level representations. In this last approach, the successive layers of the network are expected to model increasingly abstract features (Briot, Hadjeres, and Pachet, 2019)[5]. Deep music models are somehow expected to autonomously model in their first layers the musical knowledge that we explicitly provide by selecting and computing expert high-level features on the musical score.

I see two major justifications to the preference of light models:

- **Explainability:** beyond the setting-up of a predictive model, the use of high-level features facilitates our understanding of the musical phenomena that we aim at predicting (cadences, textures, structure, etc.) and favors musicological *side* findings. Such findings might occur for instance when we compare how each high-level feature contributes to the performance of the predictive model.
- **Volume of data and model complexity:** in addition to their inherent complexity, abstract musical phenomena such as cadences, textures or structures are challenging to model with supervised machine-learning methods given the generally limited amount of available annotated data. Pre-computing high-level features enables to reduce the amount of abstraction the model is expected to learn, and therefore reduces the volume of data required to train the model[6].

High-level feature approaches nevertheless require a careful selection of a representative set of features to be computed according to the current task, which can only result from a substantial musical expertise. Such approaches therefore require a crucial proximity with the musicological community, its terminology and its research on high-level musical concepts. Explicitly adding musical expertise in models also makes them necessarily dependent to some aspects of music theory and reduces their potential adaptability to alternative or future musical styles.

[4] By *light* classifier we mean a classifier which is not a deep neural network.

[5] The ability of deep neural networks to model increasingly abstract features is commonly illustrated with computer vision tasks, where modeling of shapes in the first layers allows the modeling of objects in the following ones.

[6] The preference for machine learning models that are less complex and that require less training data also fits with a growing concern in machine-learning research aiming at reducing energy consumption (Douwes, Esling, and Briot, 2021).

	corpus	features/annotations time granularity (dataset size)	features	feature example	annotation example	model
Cadence detection *(analysis)* Section 2.3.1	24 Bach fugues 42 Haydn quartets expositions	beat (B:4739 H:7173)	44	*Y-Z-bass-moves -compatible-V-I*	PAC	SVM *(label prediction)*
Texture labelling *(analysis)* Section 2.3.2	9 Mozart piano sonatas movements	bar (1164)	62	*mean number of simultaneous notes*	M1/HS1	log-reg *(label prediction)*
Form segmentation *(analysis)* Section 2.3.3	32 Mozart string quartets movements	beat (14318)	26	*triple hammer blow*	Primary theme	HMM *(label prediction)*
Function identification *(analysis)* Section 3.3.4	102 MySongBook tablatures	bar (7487)	31	*mean fret*	Rhythm guitar	LSTM *(label prediction)*
Texture imitation *(composition)* Section 3.3.5	1617 MySongBook tablatures	$\frac{1}{16}$ note (200 000)	157	*{string 2;fret 5}*	-	GRU *(content generation)*

TABLE 1.1: An overview of the tasks and associated models used in the research described in chapters 2 and 3.

It is worth noting that high-level features could possibly be used to *complete* low-level representations and make them more expressive for a number of analysis and generation tasks. While our approach in music analysis tasks rather consists in using high-level features in substitution of low-level representations, using high-level features *alone* appears in turn more complicated for generation tasks. In contrast with analysis models, generative models that are expected to generate raw musical content must indeed be trained on data including representations at this same level. Completing these representations with high-level features has shown however to be beneficial, in particular for the control of the generation (Kawai, Esling, and Harada, 2020; Makris, Agres, and Herremans, 2021).

Table 1.1 summarizes the different tasks that we have been addressing in Chapters 2 and 3, with their associated models. Although Chapter 4 describes some experiments in composer classification and end-of-phrase detection, they are not included in the table because they have not been designed for the tasks themselves, but rather to study specific aspects of some models and representations.

Evaluation Machine learning experiments commonly involve the separation of the data into a training set, a validation set and a test set, with respective size being typically around 60%, 20% and 20% of the whole dataset. The training set is used to train the model, the validation set is used to evaluate and compare different versions of the model and the test set is only used at the end to measure the best model's performance on a completely unseen subset, in order to avoid overfitting. The training set is often the largest subset, as increasing the training set size is known to improve the ability of the model to generalize to unseen data. The validation set and the train set nevertheless need to keep a reasonable size in order to be representative enough of the whole dataset and therefore provide an accurate evaluation of the model.

This trade-off is particularly critical for small datasets, as it is often the case in symbolic music processing, where each subset requires the largest possible amount of data to ensure both an efficient training and a representative evaluation. *K-fold cross-validation* consists in iterating different trainings for k different splits of the data between the training set and the validation set. The performance of the model is then provided by computing the mean and variance of the performances obtained in the

successive experiments. Although this method is only sustainable for small datasets due to the multiple trainings it requires, the final performance is presumably more representative of the model as all the data in both subsets have been evaluated. To the extreme (*i.e.*, $k = 1$), *leave-one-out cross-validation* consists in evaluating a single data on a model trained on all the others, and iterating this process on all the data.

Symbolic MIR datasets often consist in corpora of musical scores, that might themselves be split into shorter data depending on the task *e.g.*, sections, bars or beats. The high repetitiveness inherent to most tonal music generally exposes the *leave-one-out cross-validation* to potential overfitting, where the model would be evaluated on data that has previously been used for the training. This would happen at short scale for repeated patterns, and at a larger scale for repeated sections, such as the Exposition and the Recapitulation of a piece having a *Sonata form* structure.

As an alternative to *leave-one-out cross-validation*, most of the models presented in this document have been evaluated with a dedicated process we call *leave-one-piece-out cross-validation* which consists in a *k-fold cross-validation* in which each fold is associated with one song of the corpus. One major difference with standard *k-fold cross-validation* is that the size of the different folds (k) might vary depending on the length of the pieces of the corpus. One way to address this is to weight the different folds accordingly in order to keep equal importance of all data.

Chapter 2

Assisting music analysis: modeling the language of the classical repertoire

The research described in this chapter aims at modeling fundamental components of the musical language of the *Viennese classical style*, which is often represented by the instrumental music of J. Haydn, W.A. Mozart and L.v. Beethoven. The classical style is commonly associated with a set of language *elements* or *conventions*, the clearest of which is mentioned by Rosen (1997) as the short, periodic, articulated phrase. Due to the consistency of a number of these composition conventions, the classical style has become a breeding ground for corpus studies (Caplin, 2001). Fitting into this framework, this research considers that a number of composition conventions in classical music, such as *cadences* and *textures*, importantly contribute to the listener's expectation and promote the association of this style with a feeling of language.

In addition to compositional consistency, the classical repertoire is characterized by a large volume of works, involving various instrumental formations such as solo piano, string quartet or symphonic ensemble. This profusion of musical scores has contributed to encourage data-driven approaches to analyze this repertoire, necessitating the transcription of digital corpora that have a raised an important effort in the last decade in the MIR community. An important part of these scores is nowadays available in computer-interpretable formats such as Humdrum **kern, musicXML, or MEI, making possible their processing by computer programs.

This chapter opens with a brief presentation of some essential components of the classical style, followed by an overview of different computational research dedicated to this repertoire. I will then give an overview of contributions in computational classical music analysis in which I was involved. I will particularly focus on cadence modeling as a part of the PhD work of L. Feisthauer (2017 - 2021) that I have been co-supervising with F. Levé and M. Giraud, as well as texture modeling as the topic of the PhD work of L. Couturier (2021 - 2024) that I am co-supervising with F. Levé.

2.1 Elements of language in the classical style

An essential feature shared by music and natural language is their ability to induce expectation in the listener. However, while expectation in natural language results

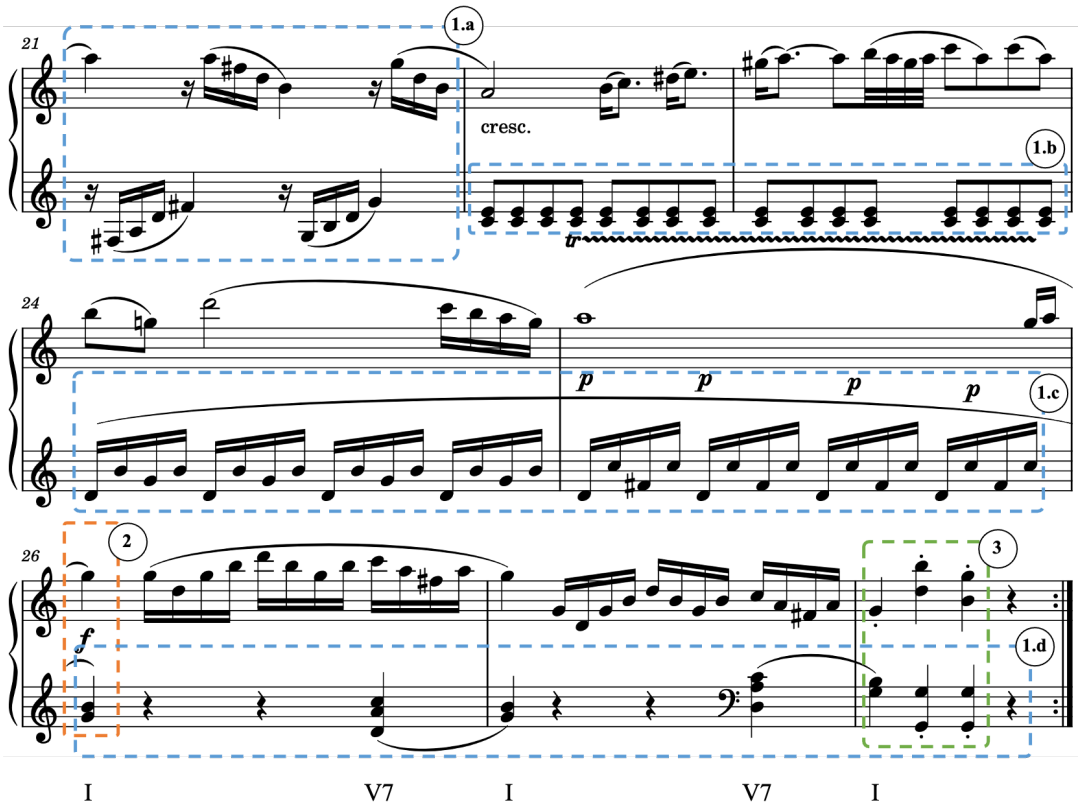


FIGURE 2.1: Last 8 bars of the exposition part of the first movement of the *Sonata facile* (Piano Sonata No. 16, K.545) by W.A. Mozart with some components typical of the classical language annotated. Textures include a *call and response* pattern between the two hands (1.a), repeated accompaniment chord referred in Section 2.3.2 as a *static* texture (1.b), an *Alberti bass* (1.c) and sparse chords (1.d). A perfect authentic cadence (2), followed by repeated V/I cadential progressions and a *triple hammer blow* at the end of the extract (3).

from semantics, expectation in music necessarily relies on alternative specific mechanisms, arguably related to the notion of musical style (Huron, 2008).

For instance, musical scores in the classical style are commonly divided into consecutive phrases. Musical phrases are characterized by a melodic segment and their underlying harmony commonly ends with a conclusive formula called *cadence* (modelled in section 2.3.1). Related to instrument configuration, *texture* (modelled in section 2.3.2) shapes the melody and the harmony in various forms, which strongly contribute to affirm the classical style of the piece. The use of keys and modulations between these keys contribute to a progressive narration. Finally, classical pieces are commonly built at a higher level following specific structures, such as the *Sonata Form* (modelled in section 2.3.3). Figure 2.1 illustrates some of these language components in an extract of a piano sonata from W.A. Mozart.

2.2 Computational modeling of classical music

The classical era left us an abundant repertoire of musical scores. Large and stylistically uniform corpora arouse a particular attention in the MIR community due to their promising ability to consistently train machine learning algorithms that aim at generalizing a musical style for unseen or generated data.

Crucial for the evaluation of models and the training of supervised machine learning algorithms, dataset contributions generally include the release of consistent score corpora accompanied by expert manual annotations of various abstract musical features. These annotations include for instance structural sections, keys, chord regions (commonly notated in relations with keys with the Roman numeral syntax), cadences or end of phrases. The creation, annotation and curation of annotated corpora has become an essential discipline within the MIR community, with particular efforts regarding classical corpora by teams such as DCML in EPFL, Lausanne.

We give here some representative examples of annotated corpora that have been used and sometimes created or extended by the Algomus team in late research. The Annotated Beethoven Corpus created by Neuwirth et al. (2018) includes harmony Roman numeral annotations for the complete set of string quartets by L.v. Beethoven, and was used for automatic harmonic analysis by Micchi, Gotham, and Giraud (2020). The Annotated Mozart Sonatas by Hentschel, Neuwirth, and Rohrmeier (2021) gathers 54 sonata movements with harmony, phrase and cadence annotations, and was used along with our texture reference annotation (Couturier, Bigo, and Levé, 2022a) for the modeling of symbolic texture as presented in Section 2.3.2. Sears et al. (2017) have released key region and cadence annotations on 50 sonata-form expositions selected from Haydn's string quartets, which we used for our research in cadence modeling presented in Section 2.3.1. The Algomus team also released structure annotations of 32 movements of string quartets by W.A. Mozart (Allegraud et al., 2019), that we used for Sonata Form modelling as presented in Section 2.3.3. We can also mention the TAVERN dataset released by Devaney et al. (2015), which includes phrase annotations that we used to compare the expressiveness of different symbolic representations in Section 4.2.2.

2.3 Contributions

2.3.1 Modeling of the classical cadence

Cadences

Musical phrases in the Western tonal repertoire often end with strong harmonic formulas called *cadences*[1]. Two phrases terminating with a cadence are illustrated in Figure 2.2. Despite this explicit structural function, cadences are defined in theoretical writings in a wide variety of ways. Based on a review of multiple music theory papers, Blombach (1987) defines the cadence as *"any musical element or combination of musical elements, including silence, that indicates relative relaxation or relative conclusion in music"*. This definition highlights the way a listener (whether musically trained or not) can *hear* the presence of a cadence by feeling that the music "breaths". A cadence

[1] from the Latin *cadere*, "to fall"

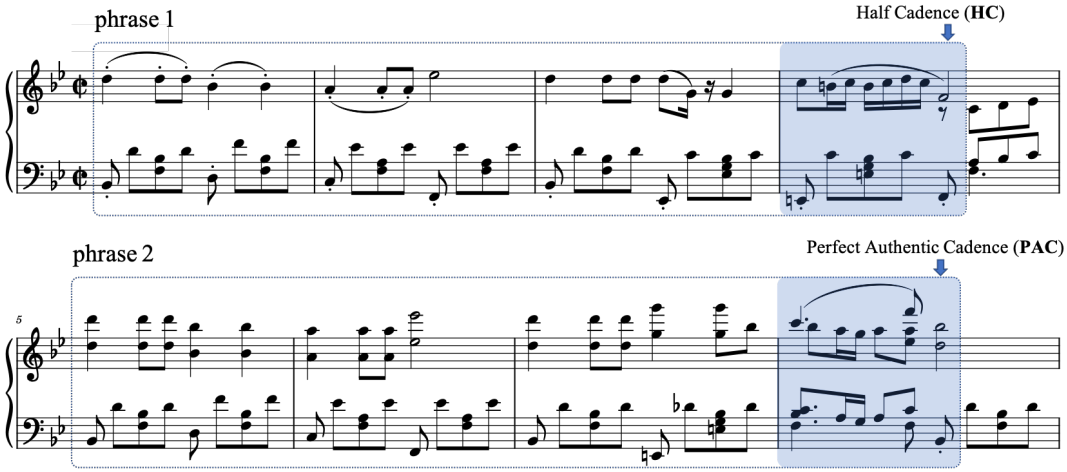


FIGURE 2.2: First two phrases of the *Impromptu No.3 (variations) in B-flat Major*, D.935, Op.142 of F. Schubert. The first four-bar phrase ends with a half-cadence (HC). The second one with a perfect authentic cadence (PAC).

is generally characterized by, possibly co-occurring, *local* musical elements, including a specific harmonic progression, a falling melody or a rhythm break. However, the (co)-occurrence of these elements do not necessarily imply a cadence.

Music theory commonly distinguishes various types of cadences classified by their underlying harmonic progression, with each progression inducing a specific feeling of closure. An *authentic* cadence (AC) is characterized by a *dominant* harmony (notated V) followed by a *tonic* harmony (notated I). In the American terminology, when both chords are in root position and the melody ends on the tonic, the authentic cadence is said to be *perfect* (PAC), otherwise it is *imperfect* (IAC). The term rIAC refers to an IAC that is in root position. *Half* cadences (HC) end with a dominant harmony, generally in root position[2]. Figure 2.2 illustrates a PAC and a HC on the first two phrases of an Impromptu of F. Schubert. Less common types of cadences include the *deceptive* cadence (DC) which is an authentic cadence where the expected final tonic is replaced by another harmony (often VI).

Sears, Caplin, and McAdams (2014) classify cadence types by *strength*, with Authentic Cadences coming first, followed by Half Cadences, then Deceptive Cadences. Notable in the frame of our parallel between music and natural language, cadences are commonly compared with punctuation in text. In particular, PACs are compared to the dot, indicating a definitive termination of the current phrase, while HCs are compared to the comma, indicating that something is terminating but that the listener is likely to expect something else to come[3].

Cadences are temporal processes that generally span over several successive beats, sometimes several bars, although they are commonly annotated in musical scores at their very last beat as shown in Figure 2.2. Using a preparation chord prior to the dominant chord, generally a *subdominant* harmony (SD, that is II, IV, or V/V),

[2] A chord is in *root position* if its root is played at the lowest pitch.

[3] Caplin (2004) however, justifiably points out that a more correct analogy with linguistic would compare cadences to *syntactical closures* rather than punctuation which consists in written signs *external* to the language.

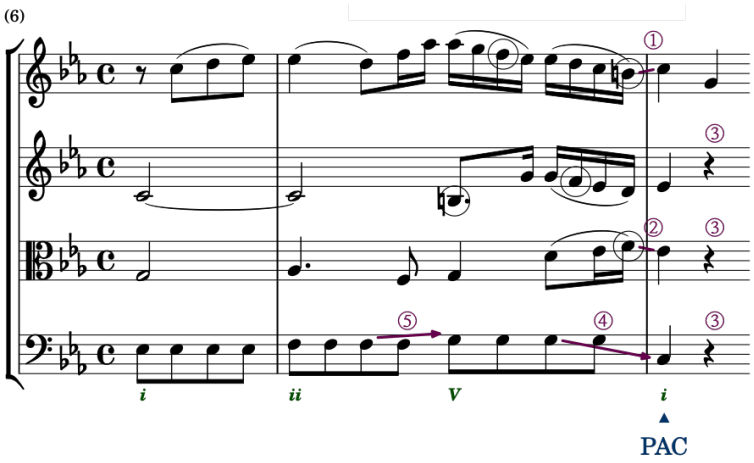


FIGURE 2.3: Haydn, op. 17/4, iv, PAC at measure 8. The high-level features computed at the beat with a PAC annotation provide information on harmony, voice-leading (①, ②, ④ and ⑤) and rhythm (③). (Bigo et al., 2018).

strengthens the salience of a PAC/rIAC. In contrast, DC and related cadences renew tension, extending the musical phrase and delaying closure until a more conclusive cadence is used.

Our work on cadence modeling has focused on Authentic Cadences and Half Cadences, which are presumably the most frequent types of cadences in the classical repertoire.

Cadence detection

This section presents our research on cadence detection (Bigo et al., 2018), which includes a part of L. Feisthauer's PhD (2018-2021) that I co-supervised with F. Levé and M. Giraud. Following the method described in Section 1.3, a set of 44 score features were identified, from music theory writings, as presumably correlated with cadence occurrence. These features were then extracted from two music corpora that includes manual annotations of cadences, and used to train an SVM model for predicting these annotations.

The selected features are computed at each beat of the score. They are boolean and can broadly be separated into categories focussing on voice-leading, harmonic and rhythmic aspects. Given the temporal nature of cadences, the computation of a majority of the 44 features requires, in addition to the present beat, information regarding its short past and future. For example, the feature *Y-Z-bass-moves-compatible-V-I* checks if the voice-leading move between the two bottom voice notes preceding the present beat fits with the typical move of a IV - V (sub-dominant - dominant) progression. The whole set of features is detailed in our article (Bigo et al., 2018).

Figure 2.3 illustrates a Perfect Authentic Cadence (PAC) annotated at the first beat of the eighth bar of Haydn's string quartet op. 17/4, iv. Among other properties, the 44 computed features explicitly encode the following knowledge:

	pieces	voices	beats	PAC (final)	rIAC	HC
Haydn string quartets	42 expositions	4	7173	99 (21)	8	70
Bach fugues	24 fugues	2 to 5	4739	63 (23)	24	5

TABLE 2.1: Description of the two annotated corpora used to train and evaluate our cadence detection model. Annotations from the Haydn string quartet corpus have been done by Sears et al. (2017). Annotations from the Bach fugues corpus have been done by Giraud et al. (2015). The numbers show that cadences are labeled at about 2% of the beats, resulting in a highly unbalanced dataset (Bigo et al., 2018).

- **Harmonic features**: the set of notes sounding at this beat form a perfect triad (C minor in this case) and the pitch-class of the highest note (C) is the tonic of the triad.
- **Voice-leading features**: the notes preceding this triad induce a $7 \rightarrow 1$ move at the upper voice (see ① in Figure 2.3), a $4 \rightarrow 3$ move in an intermediate voice (②), and a $5 \rightarrow 1$ move in the bottom voice (④). On the bottom voice, the last distinct note (F) induces a $4 \rightarrow 5 \rightarrow 1$ move (⑤).
- **Rhythmic feature**: the beat is a strong beat and it is followed by a rest at the bottom voice and at an intermediate voice (③).

Table 2.1 shows the two corpora which were used in this study. The Bach fugue corpus includes the 24 fugues of the first book of the Well-Tempered-Clavier by J.-S. Bach in which cadences have been annotated by Giraud et al. (2015). The Haydn string quartets corpus includes 42 expositions from movements of J. Haydn string quartets in sonata form, with cadences annotated by Sears et al. (2017). Even if these annotated corpora model cadences in the light of a global analysis of the form, we have used them as a benchmark on our local feature-based detection task. Note that only a minority of annotated PAC are *final* in the sense that they are included in the last four measures of the piece (or of the exposition), which makes their detection trivial.

As shown in Table 2.1, cadences are rare (less than 2% of the beats) which contributes to make their annotation delicate, in addition to the solid musical expertise generally required to spot them within musical scores. The limited availability of cadence expert reference annotation shows how crucial is the high-level feature approach for this task. By representing beats by expert high-level features, we aim at helping the model to handle first levels of abstraction and improve its capacity of assimilating the concept of cadence.

Model explainability We provide in Table 2.2 some statistics of a selected subset of features, among the total set of 44, and their co-occurrence with cadence annotations in the two corpora. These statistics aim at confirming (or refuting) the correlation of theoretical features with cadences, prior to the training of a model. In addition, these statistics will presumably help the good interpretation of the evaluation results, highlighting which features contribute the most.

For example, Table 2.2 indicates that strong beats (represented by the feature *R-Z-strong-beat*) represent 40% (1920/4739) of the beats included in the fugue corpus, but

	Bach Fugues			Haydn string quartets		
	beats	PAC	rIAC	beats	PAC	HC
	4739	63	24	7173	99	70
R-Z-strong-beat	1920	**60*** / 25	**24*** / 9	3126	**98*** / 43	**70*** / 30
R-Z-sustained-note	2341	***14**** / 31	*5* / 11	2521	***1**** / 34	***8**** / 24
R-after-Z-rest-lowest	194	**15*** / 2	**13*** / 0	1130	**59*** / 15	**34*** / 11
Z-in-perfect-triad-or-sus4	2078	**62*** / 27	**20*** / 10	3434	**97*** / 47	**55*** / 33
Z-4-moves-to-3	160	2 / 2	1 / 0	340	2 / 4	**11*** / 3
Y-Z-bass-moves-2nd-Maj	880	***0**** / 11	· / 4	559	***0**** / 7	**28*** / 5
Y-Z-bass-moves-compatible-V-I	512	**62*** / 6	**23*** / 2	578	**95*** / 7	6 / 5

TABLE 2.2: Statistics on the co-occurrence of a selection of features with manual cadence annotations in the two corpora. The column beats indicates the number of beats at which the feature returns `True`. The following columns indicate the number of co-occurrences of the feature with a cadence annotation and, in small, this expected number should the feature be random and uniformly distributed across the beats (· means 0, and not significant). Significant features therefore appear in case of large gaps between these two numbers.
(Bigo et al., 2018)

include 95% (63/60) of the PAC beats, which suggests an important, although not sufficient, contribution of this feature for the task of cadence detection. In contrast, cadences rarely involve sustained notes, interestingly making this feature negatively correlated to cadences. As expected, the presence of a rest after the beat (feature *R-after-Z-rest-lowest*, illustrated in Figure 2.3 (③)) and a 5 → 1 move in the bottom voice (feature *Y-Z-bass-moves-compatible-V-I*, ④ in Figure 2.3) are highly correlated with the presence of PAC. Contrasting with our intuition, suspended chords (feature *Z-4-moves-to-3*) were suprisingly rare in the corpus, not correlated with PACs but with HCs. As another example, the feature *Y-Z-bass-2nd-Maj* is interestingly positively correlated with HCs while negatively correlated with PACs.

Evaluation Following the method presented in Section 1.3, a classifier[4] is trained to predict at each beat of the training set whether it includes or not a cadence annotation given the 44 corresponding computed feature values. The results are indicated in Table 2.3. They show a promising ability of the model to detect PAC although detecting other types of cadences, such as HC and rIAC appears more challenging. Statistics on expert features illustrated in Table 2.2, indicate which of them contribute predominantly to the success of the prediction. For instance, the fifth bass move is the most contributing feature for the detection of PACs. HCs however, seem to lack such a characteristic move.

2.3.2 Modeling classical symbolic texture

Musical analysis in the western repertoire often involves in the first place analyzing melody and harmony in a musical score. In contrast, *symbolic texture* gathers most of the remaining facets of the score content. For instance, two accompaniment parts can follow the same chord progression but with a different texture if notes are played

[4] A linear SVM (Support Vector Machine) provided the best results as detailed in (Bigo et al., 2018).

		beats	ref	TP	FP	FN	F_1
Haydn quartets corpus (21 quartets)	PAC	3583	51	42	28	9	0.69
	HC	3583	32	18	73	14	0.29
Bach fugues corpus (12 fugues)	PAC	2357	36	26	3	10	0.80
	PAC+rIAC	2357	46	30	12	16	0.68

TABLE 2.3: Detection of cadences on the test sets of both corpora using all features: Number of beats annotated in the reference annotation (ref), true positives (TP), false positives (FP), false negatives (FN), and F_1 measure (harmonic mean of the recall and the precision). (Bigo et al., 2018)

by block (homorhythmic chord), or one by one (arpeggiated). Although intuitive for the musician, this concept is rarely formalized in comparison to other score aspects such as harmony. In this research, we define symbolic texture as the organization of notes, chords and voices within the musical score. Symbolic texture is sometimes referred as *compositional texture*. It is clearly distinct from *audio texture* that commonly refers to timbre aspects in signal processing.

The PhD work of L. Couturier, which I am co-supervising with F. Levé, aims at elaborating computational methods and model symbolic texture to facilitate the use of this abstract feature in both analytical and compositional contexts. The Master internship and the first PhD year of L. Couturier led to the elaboration of an unprecedented syntax dedicated to the description of texture in the classical piano repertoire (Couturier, Bigo, and Levé, 2022b) as well as a corpus of 9 Mozart piano movements with texture annotation at each bar and an evaluation of the ability of a set of 62 high-level descriptors to predict the texture of a musical bar (Couturier, Bigo, and Levé, 2022a).

A syntax dedicated to classical piano music symbolic texture

The piano sonata excerpt illustrated in Figure 2.1 includes various examples of symbolic textures that are typical of the classical style. The first bar shows an arpeggio pattern which is played by the two hands in a complementary way, sometimes referred as *call and response* (1.a). While the right hand is then playing a melody until the end of the excerpt, the accompaniment part played by the left hand features various textures including a repeated chord (1.b), an *Alberti bass* (1.c) and sparse chords (1.d).

These textural patterns are frequent and somehow part of a basic vocabulary of the classical style. They arguably play an essential role in the way we listen, read, play and compose music. In particular, instrumental exercises often make use of repeated patterns in a given texture. In the classical repertoire, these textural patterns are often merged, combined and declined in a variety of ways, making their formal description and computational modeling challenging in comparison to other analytical tasks such as harmony analysis. The syntax elaborated in the PhD work of L. Couturier addresses this challenge by identifying a set of musical principles of variable abstractness, whose combination can potentially describe any textural configuration found in the classical piano repertoire.

- **Layers** group notes into textural parts mainly based on onset synchrony and analogous motions. A typical 2-layers configuration would involve a melodic layer and an accompaniment layer (sometimes described as *homophonic*). Fugue forms generally feature more than two layers. The number of distinct layers is an indicator of *diversity* in a score excerpt.
- **Voices** separate simultaneous notes of a same layer. The number of voices is an indicator of the *vertical density* in a score excerpt.
- Three possible musical **functions** enable to attribute a musical role to each layer: melodic, harmonic and static. These functions are frequently combined.
- **Voice relationships** enable to indicate that two voices are played in homorhythmy, parallel or octave motions.
- **Characteristic musical figures** include various note property including sustain, repetition, sparsity, scales and oscillations.

As mentioned in Section 1.3, describing the evolution of an aspect of the musical flow, as it is the case with symbolic texture, commonly requires to carefully choose a *time granularity* indicating the frequency at which the description needs to be updated *e.g.*, every millisecond, every beat, every bar, every section, etc. Describing symbolic texture at each bar was identified as an optimal compromise. On the one hand, shorter time spans, such as beat frames, would be restrictive for the representation of the high level textural concepts previously described. On the other hand, notating the texture of several consecutive bars by a single annotation might yield to labels that would be overloaded in text and too approximate regarding the musical surface. Figure 2.4 illustrates the proposed syntax with the first bars of a piano sonata of W.A. Mozart. The framed bar has the label `3h[M1/H2h],_` because it includes 3 homorhythmic voices (`3h`) that are grouped into two layers. The first layer has a melodic function and includes one voice (`M1`) while the second layer has a harmonic function and includes 2 voices, themselves played in homorhythmy (`H2h`). The annotation finally indicates the sparsity of the texture (`,_`).

The formalization of the syntax in Backus Naur Form as well as the construction of a bestiary of typical textural configurations representative of the classical style are detailed in our publication (Couturier, Bigo, and Levé, 2022b).

A dataset of texture annotations

To support computational approaches to symbolic texture modeling, we annotated with L. Couturier and F. Levé the texture of each bar of 9 movements of Mozart Piano Sonatas with this syntax, totaling a set of 1164 annotated measures. The selected pieces were all composed in the end of the year 1774, which presumes reasonable chances of style consistency and limits the shift of compositional practice that could be induced by the evolution of piano manufacture. Moreover, these movements present an interesting diversity in tonality, rhythmic signature and tempo, while covering a large variety of textures. All those movements are in Sonata Form, which opens perspectives to pursue research on the links between texture and form as initiated by Tenkanen and Gualda (2008).

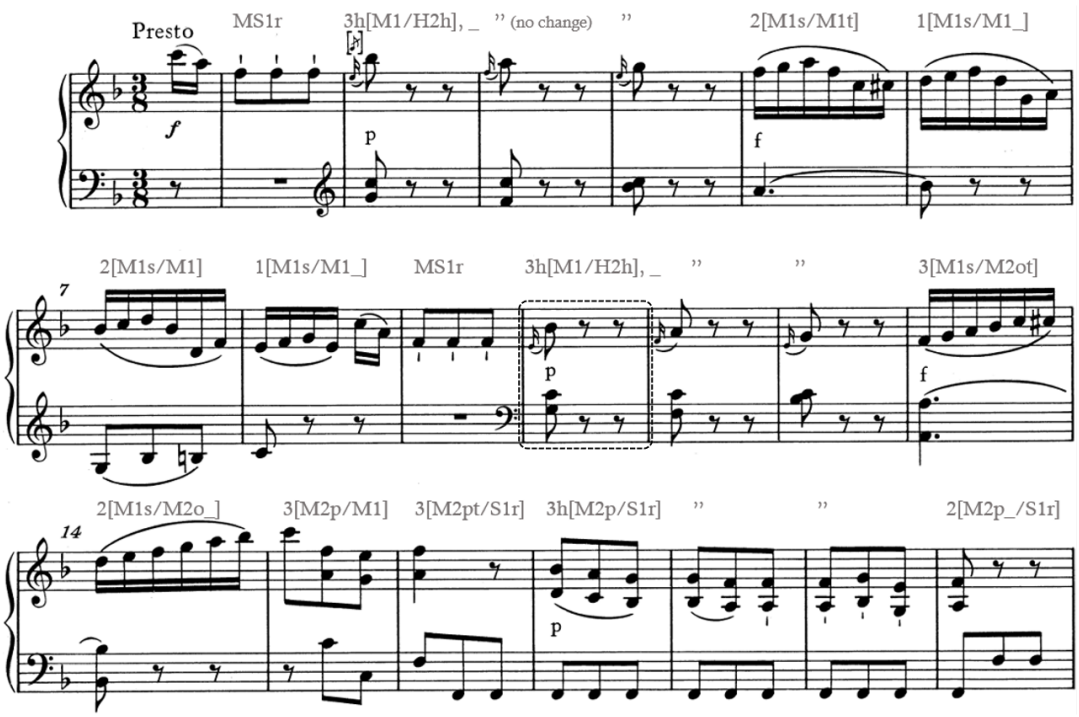


FIGURE 2.4: First bars of the *Presto* from Piano Sonata n°2 by W.A. Mozart (K. 280/189^e), annotated with texture labels following the proposed syntax (Couturier, Bigo, and Levé, 2022b).

The texture annotations were added on the musical score and saved with the web platform Dezrann. The annotations were reviewed by two musical experts who possibly provided some feedbacks to the annotations. Each feedback induced a discussion phase until a consensus between the annotator and the reviewer was reached.

The high expressiveness of the syntax facilitates the musical interpretation of large scale statistics computed on the whole annotation corpus, thus refining our knowledge on compositional practice of this repertoire by confirming or refuting empirical hypothesis and discovering unexpected properties. We can observe for instance that the most common combination in the corpus gathers a melodic layer (`M1`) accompanied by a harmony+static layer (`HS1`) as it is the case in bars 2 to 5 in Figure 2.1, confirming a common schema involving a right-hand melody accompanied by a left-hand chord realized in a repeated or *Alberti bass* form. Less expected, statistics show that the proportion of measures with harmonic layers (`H`) varies between annotated movements, notably according to the tempo: this percentage is 26% higher in slower movements (the second of each full sonata) than in the average of the others, which presumably opens promising research perspectives linking texture to style and forms.

Figure 2.5 illustrates a distribution of textural annotations depending on their diversity (number of layers) and vertical density (number of voices), providing an original overview of Mozart's piano texture practice at a high level (voices and layers). Some regions of the distribution can interestingly be associated with common musical strategies including standard polyphony (each voice is a distinct layer), homophony (one melodic layer + one accompaniment layer, see Figure 2.1, measures

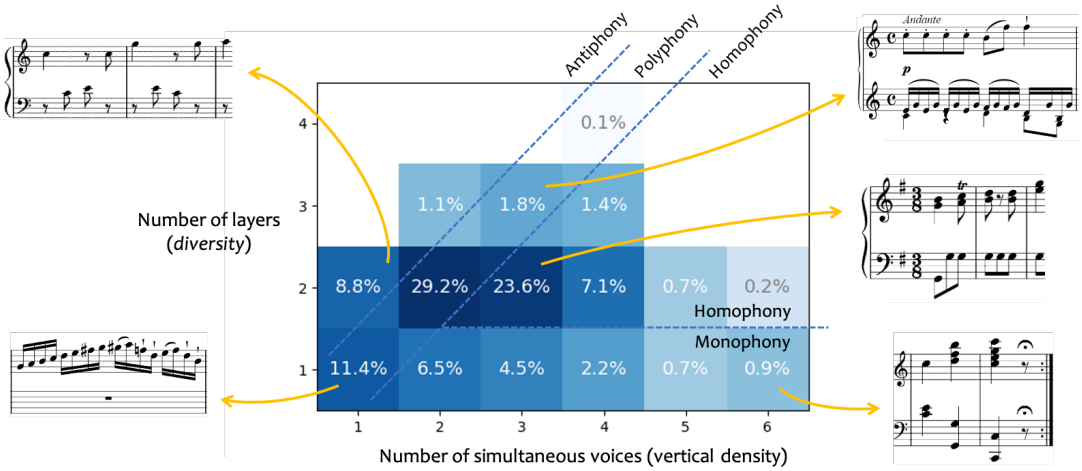


FIGURE 2.5: Distribution of textural configurations of the dataset according to their density and diversity (Couturier, Bigo, and Levé, 2022a).

2-5), monophony (one unique layer including several voices *e.g.*, homorhythmic cadences, see Figure 2.1, last measure) or antiphony, which occasionally occurs when some layers alternate (*e.g.*, call and response, see Figure 2.1, first bar).

Although the number of layers rarely exceeds 2, this distribution shows that a systematic separation into two layers, which follows an intuitive organization of the score content between the two hands of the pianist, would not be sufficiently representative of the corpus. This distribution is although presumed to be strongly related to the corpus style. For instance, multiple voice inventions, as found in the repertoire of J.-S. Bach, would presumably increase the proportion of textural configurations in the upper part of the graph, where the number of layers exceeds 2. Such reflections open the door to promising texture studies extended to other musical styles.

Generalizing the idea of organizing textural configurations in a dedicated space, which is partially experimented in the representation of Figure 2.5, opens perspectives to study texture evolution across a piece as a trajectory that could hopefully serve style and form analysis.

Towards texture prediction

In spite of a formal description provided by the syntax we proposed in (Couturier, Bigo, and Levé, 2022b), annotating symbolic texture showed to frequently involve ambiguous cases for the reviewer. This challenge motivates the design and evaluation of algorithms to automatically predict texture. In this part of the work of L. Couturier, a set of 62 score features is proposed to train a machine-learning model for texture prediction, following the general method presented in 1.3. Selected features concern pitches, onsets and slices of notes. They are systematically computed at each bar of musical scores. Using the annotated dataset presented in Section 2.3.2, a logistic regression model was trained to predict the annotation of a bar represented by its set of features. Among other findings, the results (Couturier, Bigo, and Levé, 2022a) show that the melodic function is easier to detect than the harmonic and static

functions. Homorhythmy, including parallel and octave motions, also provides fair detection results. The prediction of rare elements, such as oscillations, could be improved by increasing the size of annotated data and therefore the efficiency of the training. On the other hand, scale motives showed to require a more consistent formalization in the syntax as they happen to describe various possible behaviors.

2.3.3 Other contributions

I am briefly presenting here additional research on which I was involved with colleagues of the Algomus team. The first two topics are part of a long-term and wide research frame of the Algomus team focusing on the modeling of the *Sonata Form*. The first task focuses on section boundary prediction (Bigo et al., 2017; Allegraud et al., 2019) and the second one on *Medial Caesura* detection (Feisthauer, Bigo, and Giraud, 2019), the latter being part of L. Feisthauer's PhD. Although not detailed in this manuscript, L. Feisthauer's PhD also included a substantial work on modulation modeling in the classical style (Feisthauer et al., 2020).

Sonata form: section boundary identification

The *Sonata Form* refers to a musical structure that has been widely used in the classical repertoire since the middle of the 18th century. Musical pieces in the sonata form include three successive large-scale parts respectively called *Exposition*, *Development* and *Recapitulation* that include a characteristic succession of tonalities. A *Primary theme*, a *Transition*, a *Secondary theme* and a *Conclusion* are consecutively used in both the Exposition and the Recapitulation. Importantly, the secondary theme of the Exposition is played in another key, frequently "at the dominant", meaning that the keys of the primary and secondary theme are distant from a fifth interval. In contrast, both themes are played in the same key in the Recapitulation. A half-cadence, called *Medial Caesura*, generally takes place right before the beginning of the secondary theme (Hepokoski and Darcy, 2006). Figure 2.6 illustrates the successive sonata form's sections annotated on the score of a string quartet movement by W.A. Mozart.

Following the method presented in Section 1.3, a set of high-level score features was selected for the task of section boundary detection. The features alternatively combine melody, harmony, and rhythm aspects. Some of them are strongly specific to structure in the classical style. For instance, the *triple hammer blow* consists in a flagrant repetition of a chord, generally the resolution chord of a cadence, to emphasize the end of a section. The last bar in Figure 2.1 and the second bar in Figure 2.7 both include a triple hammer blow. Another characteristic feature is the *rhythm break* which aims at detecting the interruption of repetitive rhythms that consist in at least 15 consecutive notes with same duration, which is likely to occur at the end of a Transition section, right before the Medial Caesura. Figure 2.7 illustrates the detection of these features in an excerpt located at the Medial Caesura of a string quartet movement exposition.

These features were used as observations of a Hidden Markov Model predicting boundary locations. Transition and emission probabilities were hard coded from theory-driven intuitions in a preliminary version of the model (Bigo et al., 2017) and learned on an annotated corpus in a more extended version (Allegraud et al., 2019).

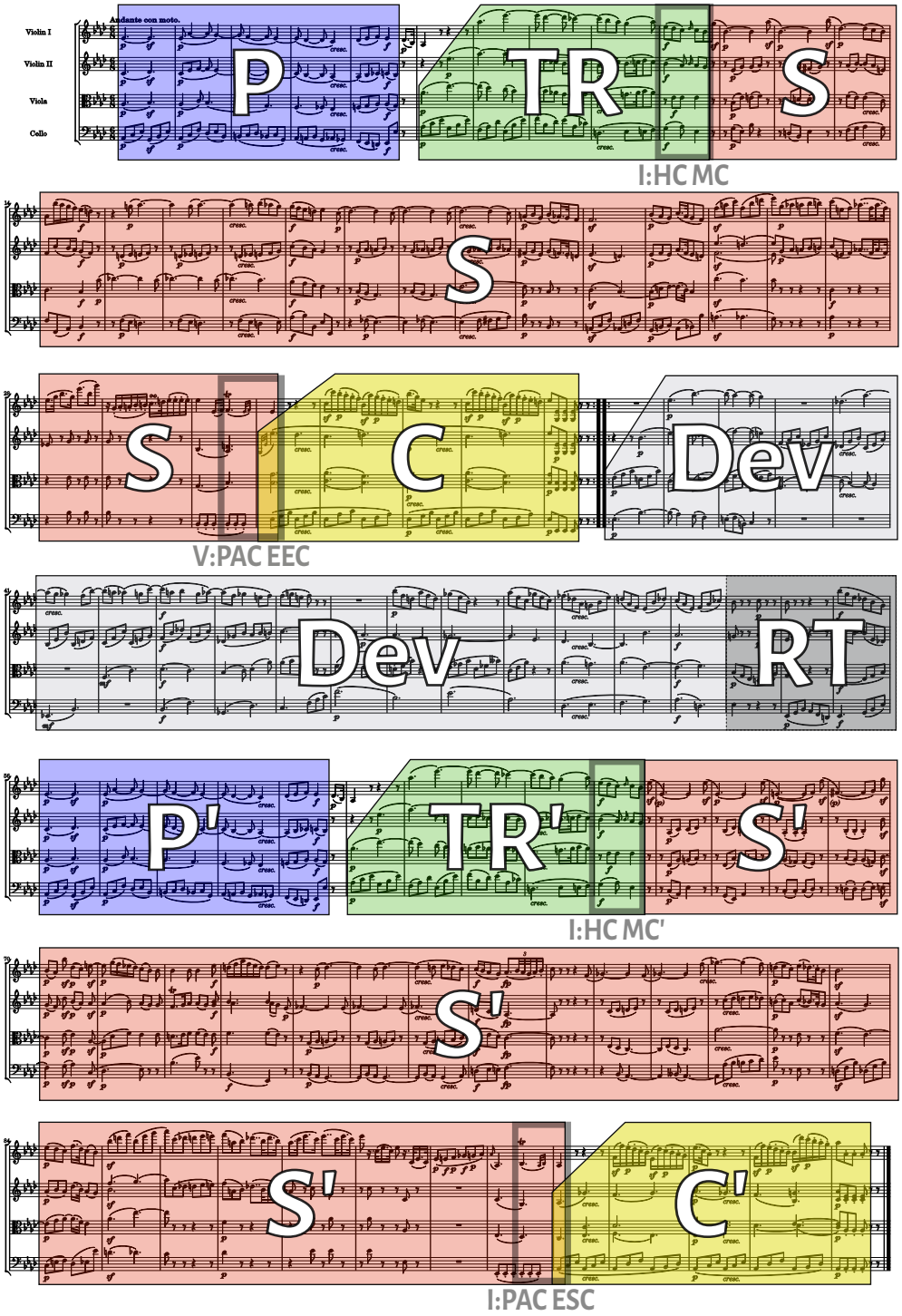


FIGURE 2.6: *Andante con moto* of the String Quartet #16 in E♭ Major, K. 428, 2nd movement from W.A. Mozart following a sonata form. Following notations of Hepokoski and Darcy (2006), the primary themes (P/P') are followed by transitions (TR/TR'), ended with Medial Caesuras (MC/MC') – they are here Half Cadences (HC) in the main tonality (I). In the exposition, the secondary theme (S) and the conclusion (C) are here in the tonality of the dominant (V, B♭ major). In the recapitulation, both S' and C' come back to the main tonality. Between the exposition and the recapitulation, the development (Dev) moves to other keys and is concluded by a re-transition (RT) focusing on the dominant of the primary key (Allegraud et al., 2019).

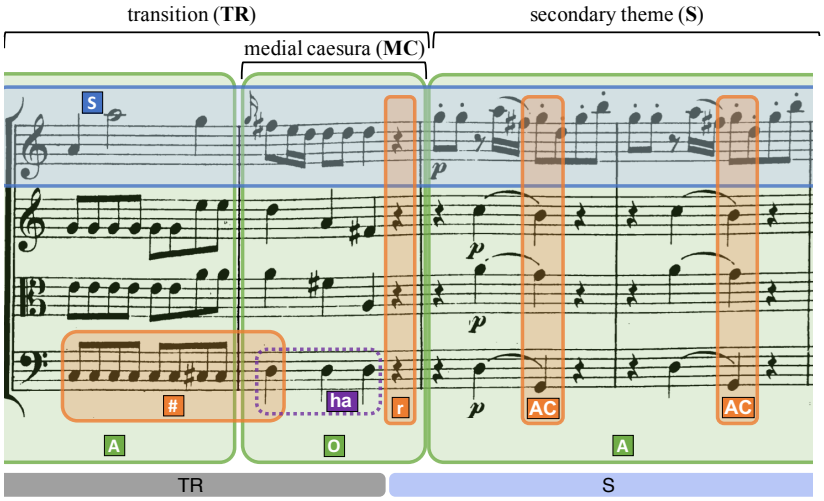


FIGURE 2.7: First medial caesura (MC) in the first movement of the String Quartet no. 4 in C major by W.A. Mozart (K. 157), measures 29 to 32. The MC ends the transition (TR) and is before the beginning of the secondary theme (S). Several high-level features are computed within this region: the thematic pattern *S* (prematurely detected as it is supposed to begin after the MC), tonality regions (*A* and a wrong additional one *O*), a chromatic upward bass movement *#* and a full rest *r*. Two authentic cadences *AC* are also wrongly detected at the beginning of the secondary theme. A *triple hammer blow* (*ha*) characteristic of the medial caesura is also detected (Bigo et al., 2017).

This last work publicly released our annotation dataset of Sonata Form structures including 32 movements of string quartets by W.A. Mozart.

Figure 2.8 compares predicted and reference structures for six pieces of the dataset, highlighting abilities of the model to predict section boundaries on simple cases (as K. 172.2) but also important difficulties in more challenging pieces (K. 465.1). The limits of the model are presumed to be caused mostly by the small size of the corpus, exacerbated by an important diversity of section combinations in sonata forms induced for instance by occasional inclusion of Introduction and Coda sections.

Sonata form: *Medial Caesura* detection

As a key component of the Sonata Form, the *Medial Caesura* has been the topic of numerous musicological studies aiming at exploring its nature and function. Hepokoski and Darcy (1997) explain that the Medial Caesura closes the first part of the Exposition, concluding a process of energy gain initiated during the Transition section, and makes the second part available thanks to that energy accumulated.

The Medial Caesura modeling is particularly challenging given its musical abstractness[5]. As part of his PhD, L. Feisthauer proposed and implemented a set of 13 beat high-level theory-driven features essentially relating to pitch and rhythm for the detection of Medial Caesura (Feisthauer, Bigo, and Giraud, 2019). These features were used to describe the current beat and its short-surrounding, and eventually

[5]The Medial Caesura is probably the most abstract concept that was attempted to be modelled in the Algomus team.

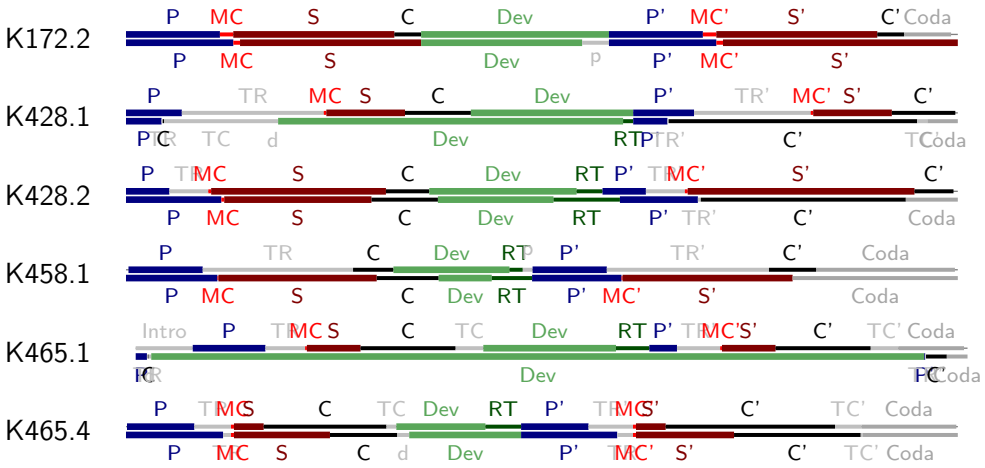


FIGURE 2.8: Comparison between the reference analysis (top) and the predicted analysis (bottom) on six string quartets movements (Allegraud et al., 2019).

train a recurrent neural network (Long-Short-Term-Memory) for the task of predicting the occurrence of a Medial Caesura at each beat.

In contrast with the cadence detection task described in Section 2.3.1, only features describing the present beat were selected here. For this reason, it was chosen to use a recurrent model to capture the progressing aspect of the Medial Caesura. An LSTM was trained in a *Leave-One-Piece-Out* validation framework on a corpus of 27 two-part expositions of string quartets from W.A. Mozart included in the corpus used for Sonata Form detection described in Section 2.3.3.

Figure 2.9 displays the probability of the successive beats of four pieces to be a Medial Caesura as estimated by the neural network. Probability curves facilitate the comparison of the ambiguity of the pieces for the Medial Caesura detection task, and provide a rate of wrongness for prediction mistakes.

2.4 Impacts

The research of this chapter, and more generally of the Algomus team, is probably located in the most musicological part of the MIR community. While a number of MIR models dedicated to the analysis of classical music focus on a set of general tasks including key detection, harmony analysis, or composer classification, a large part of our work aim at modeling slightly more specific musical phenomena such as symbolic textures, cadences, and, as a more extreme case, medial caesuras. One hope regarding this specific positioning is to raise musicologists interest in computational approaches, in particular the music analysis community who still seems under-represented in the MIR community.

As mentioned in Section 1.2.1, predictive algorithms as those presented in this chapter have few chances to be directly used by musicologists. As proposed by Dahlig-Turek et al. (2012), this might be partly explained by shortcomings often characterizing such tools including their specificity to a certain repertoire or approach, their lack of robustness and flexibility, their imperfect user interface or the complexity of their

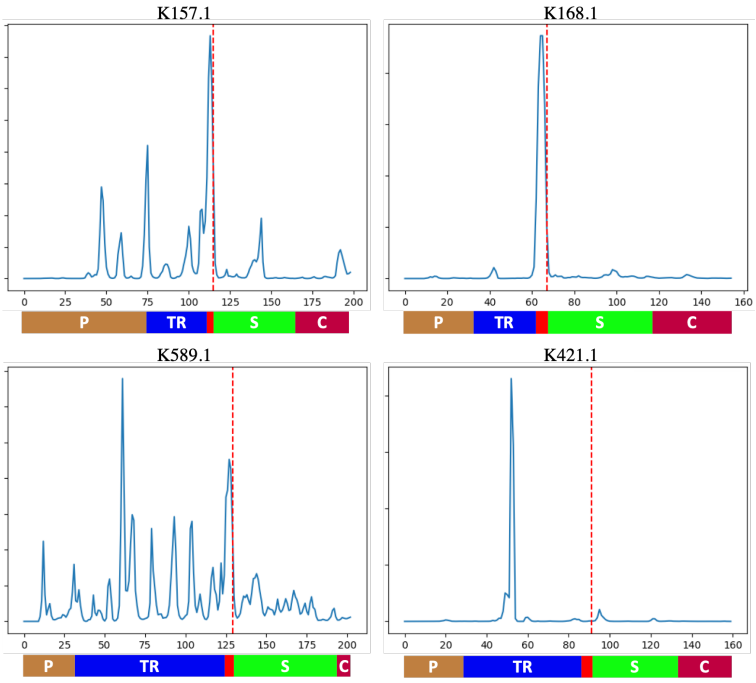


FIGURE 2.9: Estimated probability of Medial Caesura in four pieces. The probability is computed for each beat of the score. On the top: two correct predictions, including a particularly unambiguous piece (right). On the bottom: two wrong predictions, the right one being more frankly wrong (Feisthauer, Bigo, and Giraud, 2019).

output. Nevertheless, the elaboration of these tools in close collaboration with the musicological community contributes to diversify viewpoints to discuss common interests. Favoring explainable models as described in Section 1.3, allowing statistical observations such as those on cadences illustrated in Table 2.2, is a way to foster such interdisciplinary collaborations. Although our publications on the topic still haven't been significantly cited in musicological publications to our knowledge, we aim at maximizing opportunities of research sharing with musicologists. On the topic of harmony, a part of the PhD work of L. Feisthauer has for instance been presented at a musicology conference dedicated to the 300th anniversary of J.-P. Rameau's *Traité de l'harmonie*[6] to provide a computational point of view on research in this topic. The PhD work of L. Couturier on texture enables to pursue a collaboration of the Algomus team with French musicologist N. Hérold, in the same way the Algomus team did in the past with M. Rigaudière. The research on guitar tablature presented in Section 3 is also done in close collaboration with musicologist B. Navarret. Bridging this gap is however challenging due to a number of notable differences between our communities. For instance, methodologies, problem descriptions and the frequent quest of generalizing models by computer scientists might appear reductive and over-simplified for musicologists who frequently have more subtle knowledge on particular cases of the repertoire.

The interest brought by our results seems however more visible in the MIR community in which our work on cadence detection has lately been referred as the state

[6] https://www.iremus.cnrs.fr/fr/evenements/1722-2022-trois-siecles-du-traite\-de-jean-philippe-rameau-la-musique-science-devant-la

of the art by Karystinaios and Widmer (2022). Among other citations, this work is also mentioned by Hentschel, Neuwirth, and Rohrmeier (2021) as a motivation for the building of consistent corpora dedicated to the modeling of high-level components of the classical style.

Although part of a more long-term vision, computational models might finally contribute to music pedagogy if they are embedded within digital score edition or visualization software. The Dezrann platform for instance aims in the future to provide experimental models to its users for specific tasks such as cadence detection on any uploaded musical score. Moreover, the idea of systematically describing musical score regions (beats, bars, sections) with high-level features potentially facilitates the search of specific musical excerpts in a repertoire, respecting a precise set of constraints intended to illustrate musical situations in a pedagogical context.

2.5 Perspectives

Most of my research perspectives on computational classical music analysis are linked to the modeling of texture in the frame of the PhD of L. Couturier that I am co-supervising with F. Levé. This project will include the identification of methods to organize distinct symbolic textures in a topological space based on their similarity. Possible approaches to estimate texture distance include text distances between texture labels. Texture distances will also be studied empirically by conducting perceptive listening tests asking human subjects to estimate perceived distances between musical examples featuring representative textures. We also plan to evaluate geometric distances between explicit feature vectors or latent vectors computed by unsupervised models.

The research presented in this section are limited to supervised machine learning experiments in which a model is trained thanks to a set of expert reference knowledge. In contrast, unsupervised machine learning approaches do not require any expert annotations, and try to model abstract knowledge from unlabeled data. While presumably more challenging to model abstract phenomena, unsupervised approaches open the perspective of modeling the classical style in a more generic way with the potential discovery of essential stylistic patterns less formalized than essential component of music theory such as cadences. This seems particularly true for polymorphic features such as texture whose exhaustive formalization is complicated by endless combination possibilities. The modeling of texture with unsupervised approaches is therefore an essential perspective of this work.

Textural spaces can be used to study the evolution of the texture of a musical piece as a trajectory, which would bring an original approach to model musical style. The work of L. Couturier also intends to bring contributions in the field of style transfer in which a musical sequence is transformed to fit with the musical style of another one (Cífka, Şimşekli, and Richard, 2020). Being a major component of musical style, the explicit modeling of symbolic texture is felt to potentially bring significant improvement in this field. Future works in texture will also involve methods of music generation guided by texture scenarios, which seems to have been approached only with very basic textural descriptors to date (Makris, Agres, and Herremans, 2021).

The predominant use of deep neural networks in symbolic MIR tasks nowadays also encourages us to study the ability of these models to model textures across their successive layers. The field of image processing appears as a promising source of inspiration and analogies for this task given the ability of convolutional networks to model visual concepts with progressive levels of abstractions (pixels, textures, shapes, objects, etc.).

We hope that modeling musical texture could ultimately help larger scale tasks in music analysis including structure detection. The Sonata Form modeling described in Section 2.3.3 could indeed be improved by adding some textural features that are felt to be highly informative regarding changes of sections as foreseen by Tenkanen and Gualda (2008). Texture modeling could probably also help more general tasks such as harmony analysis. For short-term phenomena such as harmony however, our texture modeling approach would require to be more modular regarding time granularity, as the model presented in Section 2.3.2 only supports texture modeling at the level of a score bar. We also believe this work can bring intuitions far beyond the frame of the classical style. While the analysis of harmony is often restricted to Western tonal music, textural concepts are prone to describe a much wider range of musical cultures and styles. The Section 3.3.5 describes some research aiming at imitating the texture of rhythm guitar sections in the modern popular style, which will hopefully benefit from findings in classical piano texture modeling.

Chapter 3

Assisting music composition: modeling the language of guitar tablatures

The guitar is today one of the most played instruments in the world (Trades, 2021) with remarkably diversified practices and repertoires. Many guitarists, beginners or experts, transcribe musical ideas by writing *tablatures*. Tablatures are a specific type of musical score that include gesture annotations specific to the instrument. Guitar tablatures in particular, indicate the string and the fret at which a note must be played, as well as specific playing techniques enabled by the instrument such as *bend notes*. Figure 3.1 illustrates a guitar score extract in both standard notation (top) and tablature notation (bottom).

This chapter describes research around guitar tablature modeling, motivated by the elaboration of computational tools to assist the composition of guitar music in musical styles that can be grouped under the general term of *modern popular music*. This project has been initiated in 2017 when the Algomus team met the French company Arobas Music, which edits the tablature notation software Guitar Pro and maintains the tablature corpus MySongBook. This collaboration has led to three conference articles to date. Since October 2022, I am coordinating the 4-year ANR JCJC TABASCO project on computer-assisted tablature composition. Notably, this project opens a close collaboration with musicologist B. Navarret (IReMUS, Sorbonne Université), specialist in the field of guitar practice.

I will first present some essential components of guitar tablatures, trying to highlight how they contribute to associate guitar music in the modern popular style with a specific kind of language. After reviewing some contributions in the area of guitar music processing, I will present some of our own contributions as well as future directions mostly formalized in the TABASCO project proposal.

3.1 Elements of language in guitar scores

This section describes two specific aspects of guitar practice that are assumed to contribute to establish the language of the modern popular style. We will first focus on guitar music notation, and more specifically the predisposition of the tablature system to transcribe *gestures* preformed by the guitarist. We will then highlight the ability of the guitar to play various musical functions within a modern popular orchestral formation.

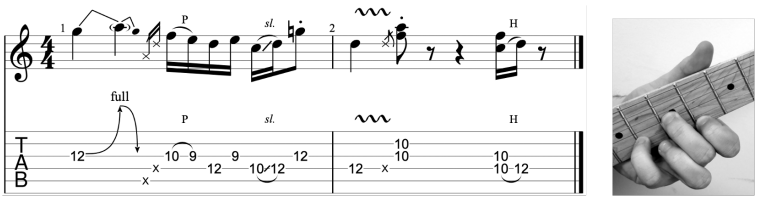


FIGURE 3.1: Score excerpt of a guitar solo from the song *Another brick in the wall* (Pink-Floyd). The top part is a standard notation. The bottom part is a tablature notation indicating string/fret combinations and some playing techniques. Playing techniques are, from left to right, a full *bend*, a *Pull-Off*, a *slide*, a *left-hand vibrato* and a *Hammer-On*. Tablature rendered with Guitar Pro.

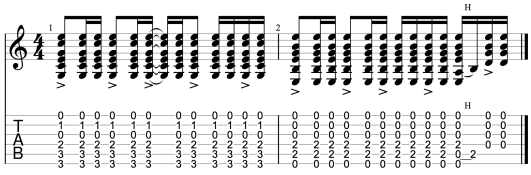
3.1.1 Gesture annotations

As with many stringed instruments, guitar scores commonly include a standard score notation upon a tablature notation as illustrated in Figure 3.1. Standard notation represents musical content in terms of notes and chords. It can potentially be used to notate music for different types of instruments. In contrast, tablature notations are generally specific to one given instrument. Navarret (2013) argues that guitar tablatures are initially intended to transcribe *performance gestures*. Gesture annotations in guitar include finger positions, indicated by *string+fret* combinations, as well as a variety of playing techniques made possible by the making of the instrument.

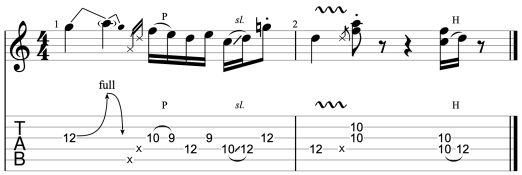
- **Position annotations**[1] Guitar tablatures include horizontal lines representing the strings of the instrument. Numbers displayed on the lines indicate the *fret*, which is a discretized level on the fretboard at which the string must be pressed. For example, the first note of the excerpt in Figure 3.1 is a G_7 which is played by pressing the fourth string at the twelfth fret. Given the tuning of the guitar, string/fret combinations enable to deduce the pitches they are producing. The reverse deduction is however not possible as a same pitch can generally be produced with several string/fret combinations. For instance, the G_7 pitch could have also been produced at the height 8th fret of the fifth string.

- **Playing techniques annotations** In addition to fretboard positions, the tablature at the bottom of Figure 3.1 includes a number of *playing techniques* annotations. Playing techniques annotations indicate specific gestures on the guitar, which contribute to improve the expressiveness of the performance. They include left-hand techniques such as *bends* that indicate that the string is progressively pulled, inducing a continuous pitch shifting, as well as other techniques such as *hammer-on/pull-off*, *slides*, *let-ring* and *left-hand vibratos*. They also include the more rarely studied right-hand techniques such as *palm-mutes*, *right-hand vibratos*, *artificial harmonics* which frequently appear in modern pop/rock tablatures.

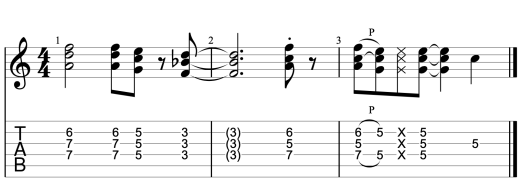
[1] Position annotations are often ambiguously referred to as *fingering annotations*. We prefer however the term *position annotation* to avoid ambiguity as *fingering* commonly refers to finger choice, including for other instruments like the piano.



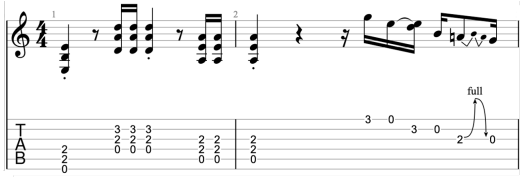
(A) Extract from a rhythm guitar section from *Space Oddity* (David Bowie)



(B) Extract from a *lead guitar* section (part of a solo) from *Another Brick In The Wall* (Pink Floyd)



(C) Extract from *Sultans of Swing* (Dire Straits) combining chords and melody.



(D) Extract from *Back In Black* (AC/DC) alternating chords and melody.

FIGURE 3.2: Four tablature extracts illustrating various degree of the rhythm guitar function. Tablatures rendered with Guitar Pro.

Notating gestures has two essential conveniences. From a transcription perspective, tablatures convey more stylistic information than the score content alone. The choice of fretboard positions as well as the use of playing techniques indeed strongly contribute to the style of a guitar part (McVicar, Fukayama, and Goto, 2015). From a performance perspective, although sight-reading tablatures requires a certain familiarity with the physical instrument, it is generally more accessible than a score as it requires less knowledge in music notation and theory. It is possible that this property contributes to encourage some music beginners to orientate towards the choice of the guitar.

3.1.2 Guitar playing functions in modern popular music

Similarly to other instruments like the piano, a guitar part in a modern popular music ensemble, especially in the pop/rock style, can potentially be associated with various musical *functions*, or *roles*, in a song. Most of the time, these functions can be gathered within two broad categories being accompaniment (commonly called *rhythm guitar*) and melody (commonly called *lead guitar*). Figure 3.2a illustrates a typical rhythm guitar part and Figure 3.2b a typical lead part. Interestingly, it is even common to distinguish two guitarists in a band, especially in rock bands, as the *lead guitarist* and the *rhythm guitarist*. Although not central in our research and less frequent in the context of a pop/rock ensemble, it is worth noting that the guitar, as the piano, can simultaneously perform accompaniment and melody. While the piano will typically split the two functions over left hand and right hand, the guitar will generally use a specific playing technique called *finger picking*[2].

[2]Jazz guitar includes some other practices to play both chords and melodies, including fast alternations between both functions.

A first level for the description of the function of a guitar section is to estimate if it is thought to be perceived in the *background* or in the *foreground* of the song. Accompaniment parts will generally fit the first category as they often aim at supporting a main musical part like a singing part or an instrumental solo. Melodic parts are in turn generally thought to be perceived in the foreground. However, it is not uncommon for a lead guitarist to play a background melody, possibly improvised, during singing sections[3].

Rhythm guitar sections in the modern popular repertoire mostly consist in sections realizing a chord sequence[4], generally in a repetitive way. In contrast, *lead guitar* appears to be less well-defined as it can be alternately associated with solo parts, as in Figure 3.2b, *riffs*, *licks*, or diverse melodic parts. In addition to these prevalent categories, guitar tablatures can possibly include arrangement parts of diverse forms, which could hardly be classified in one of the previous categories. Figure 3.2c illustrates an ambiguous case where the guitar plays chords with a texture which is presumably more common in lead guitar parts than in rhythm guitar parts. Figure 3.2d illustrates a guitar part that includes three *power chords*[5] followed by a short melodic pattern which is played at the transition between two occurrences of the chord sequence. Sections 3.3.4 and 3.3.5 focus on tablature texture modeling with applications on detection and imitation of rhythm guitar sections.

3.1.3 Guitar tablatures and composition

The principle of guitar tablature composition in the modern popular style could almost be considered as a nonsense. Indeed, Navarret (2013) outlines that for most modern popular music, the score is not a prerequisite for the creation, but rather a document resulting from the creation, essentially used for memorization and transmission. Composition in modern popular music seems indeed to mostly occur with experimental instrumental performance, but with limited use of music notation (Deruty et al., 2022). This contrasts with classical music composition in which score notation presumably occupies a much more central place, although classical composers also occasionally use higher-level representations such as sketches or blocs.

Nevertheless, the research described in this chapter is based on the hypothesis that the technological functionalities offered by modern music notation software, including playback and instruments/sound effects, commonly incite the composer to perform substantial modifications at the transcription stage. We therefore believe that the access to a judicious selection of AI-based functionalities embedded within music notation software can potentially contribute to renew composer's habits and shift some parts of the composition process at the notation stage. It is worth noting that similarly, the sophisticated functionalities of recent music production software, also known as digital audio workstations, now play a central role regarding the decisions involved in the production of modern popular music (Bell, 2018). A part of the research described in this chapter aims to encourage an analogous phenomenon

[3]Examples of this behavior include the verses of the song *What's Up* (4 Non Blondes) or the bridge of the song *Cryin'* (Aerosmith).

[4]In some cases however, the rhythm guitar can include a repetitive melodic line, as in the verses of *Always on the run* from L. Kravitz or *Give It Away* from Red Hot Chili Peppers, which both feature a single chord.

[5]*power chords* refer to chords limited to a tonic and a fifth. They are widely used in rock and metal styles, generally with distortion/overdrive sound effects.

at the level of the guitar tablature. For instance, composition assisted functionalities can propose alternative content on selected regions of the score, following recent *inpainting* methods (Hadjeres and Crestel, 2021). Such functionalities could also help the composer in completing its composition on aspects he feels less skilled, for instance an accompaniment part played by an instrument for which he feels unskilled.

The question of user interface is naturally crucial to reach composers. The elaboration of composition functionalities in notation software should therefore be addressed with close consideration of recent studies on the real use of algorithmic tools by composers (Deruty et al., 2022; Kayacik et al., 2019).

3.2 Computational processing of guitar tablatures

Most research involving computational processing of guitar tablatures fall into three categories: position prediction, style analysis and content generation.

The position prediction task, also referred as to automatic fingering or score-to-tab, results from the fact that a same note can generally be played at multiple locations on the guitar fretboard as mentioned in Section 3.1.1. This task therefore consists in estimating a string/fret combination for each note of a score in order to optimize its global playability. The fingering problem has been approached with a variety of methods including HMM from audio signal (Barbancho et al., 2011) and symbolic scores (Hori and Sagayama, 2016), and visual detection (Burns and Wanderley, 2006).

Computational guitar tablature analysis include the detection in audio recordings of playing techniques described in Section 3.1.1 (*bends*, etc.) (Reboursière et al., 2012; Chen, Su, Yang, et al., 2015). Analysis of audio guitar recordings also include automatic transcription of tablatures (Xi et al., 2018; Wiggins and Kim, 2019) based on the training of convolutional neural networks on guitar recording spectrograms, which jointly tackle pitch and fingering estimation. Computational analysis of symbolic tablatures also include guitarist style modeling with Markov models (Das, Kaneshiro, and Collins, 2018) or with directed graphs (Ferretti, 2016). Our work described in Section 3.3 includes a corpus study on fretboard position (Cournut et al., 2021), as well as a method for the identification of rhythm guitar sections (Régnier, Martin, and Bigo, 2021).

Guitar tablature generation has been approached with various methods including HMMs to generate guitar arrangements from audio polyphonic music (Ariga, Fukayama, and Goto, 2017), integer programming to generate blues solos (Cunha, Subramanian, and Herremans, 2018), and transformer neural networks to generate fingerpicking tablatures (Chen et al., 2020). Guitar tablature generation has also been limited to rhythm guitar and lead guitar (McVicar, Fukayama, and Goto, 2014b; McVicar, Fukayama, and Goto, 2014a) with probabilistic methods.

3.3 Contributions

3.3.1 Industrial collaboration with *Arobas Music*

My research in computational tablature modeling take place in the framework of an industrial collaboration with the French company Arobas Music based in Lille. Arobas Music produces the tablature edition software Guitar Pro[6] which is used by more than 300 000 users around the world. The native Guitar Pro file format (`*.gp`) has become a standard which is nowadays readable by most tablature software.

In addition to the development of Guitar Pro, Arobas Music owns the tablature corpus MySongBook which includes more than 2500 songs, mostly in the modern popular style, all transcribed by professional musicians, resulting in an unprecedented high quality. The corpus includes pieces in the pop/rock, metal, classical, and jazz styles. These pieces have been selected depending on their estimated popularity within the guitar practice community. Although heterogeneous in style, this selection is therefore supposed to reasonably reflect a large part of western guitar practice.

The collaboration between Arobas Music and Algomus began in 2017 after the participation of the company to an industrial panel discussion at the French conference *Journée d'Informatique Musicale* which was organized by Algomus in Amiens and for which I was a co-chair of the organization committee. In addition to the geographical proximity between the two teams, preliminary discussions enabled to highlight a number of common scientific and musical reflections opening the door to promising collaborations. I strongly believe that a number of feasible MIR research could potentially reach to innovative improvements of tablature software such as Guitar Pro with a high impact on their user community. These include methods to assist the notation, composition, learning, import, arrangement and playback of tablatures.

Although the MySongBook corpus is not be publicly released, Arobas Music provides to the Algomus team exclusive access to the dataset and allows the publication of findings based on it, as well as high-level representations computed from the data that can be reused in MIR research by other teams working on guitar tablature processing. Alternatively, the publicly released DadaGP corpus (Sarmento et al., 2021) includes 25 000 songs in the `.gp` format, resulting from diverse contributions of the guitarist community. Although much larger, this corpus includes tablatures transcribed with a much lower quality.

The research done so far with the MySongBook corpus has resulted in the modeling of musical knowledge at different abstraction levels. Section 3.3.2 details the development of vector representations, referred to as *encodings*, to manipulate `.gp` tablatures at a low level and facilitate their processing by machine learning models. Section 3.3.3 presents results of a statistical study of the corpus focusing on string/fret positions occurrences across different styles. Section 3.3.4 presents a method for the automatic identification of the *musical function* of a tablature excerpt. Section 3.3.5 presents ongoing works on imitation of tablature texture, intended to assist rhythm guitar composition. Although not directly related to composition, the first works all constitute building blocks on which the latest is built on.

[6] https://www.guitar-pro.com/

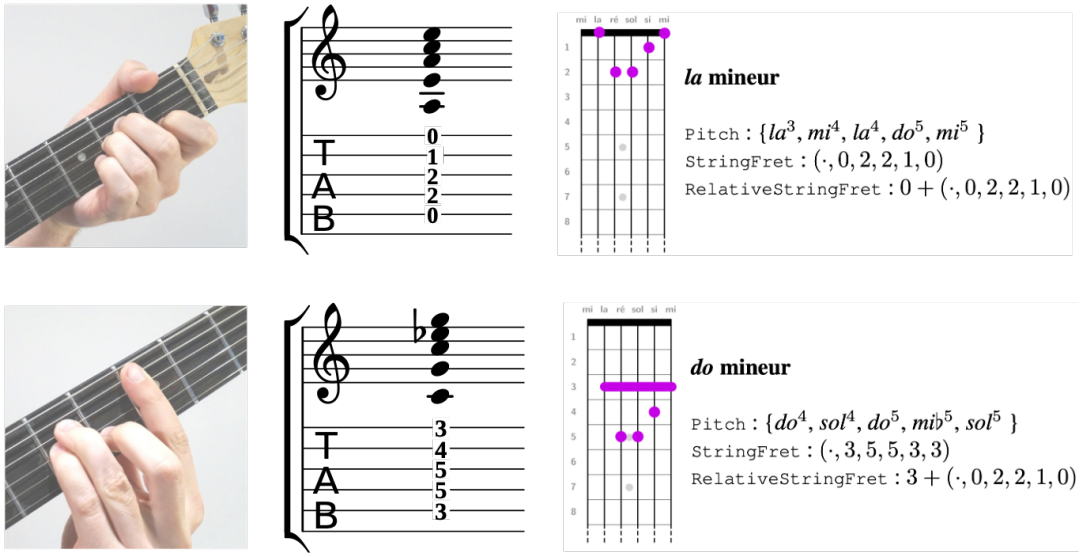


FIGURE 3.3: Two guitar positions (left) with their tablature notation (middle) and their encodings (right) (Cournut et al., 2020).

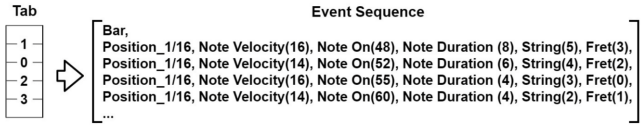


FIGURE 3.4: Token encoding of guitar tablature proposed by Chen et al. (2020).

3.3.2 Tablature encodings

This section describes the work done by J. Cournut during his internship followed by a three months contract, that I co-supervised with M. Giraud and which reached to two publications (Cournut et al., 2020; Cournut et al., 2021).

Aiming to facilitate the processing of symbolic tablatures in the .gp format, the first achievement in this project was the implementation of a music21 parser transforming a .gp format tablature, structurally equivalent to XML, into a well-formed music21 python object. Several encoding methods were also elaborated for the systematic representation of music21 tablatures as binary vectors compatible with machine learning algorithms. In the context of machine learning experiments, the musical flow is indeed commonly encoded as sequences of binary vectors, which transcribe which pitches are played at the successive time steps of the musical score as reviewed by Briot, Hadjeres, and Pachet (2019). In addition to usual note information such as pitch and duration, tablatures require the encoding of gesture information described in Section 3.1.1. Figure 3.3 illustrates three encodings of two common guitar positions. For the sake of clarity, fret positions are indicated with their decimal values, although the real vectors only include binary values (Cournut et al., 2020).

- the `Pitch` encoding transcribes at a given time step the set of sounding pitches among the 49 pitches available on a standard guitar fretboard. It is equivalent

to the *piano-roll* representation, which is generic to any instrument and does not include any position information.

- the `StringFret` encoding transcribes position information with 150 values, associated with the complete set of string/fret combinations on a standard guitar. Note that the `Pitch` encoding can be deduced from this one, assuming the tuning of the guitar is known.
- the `RelativeStingFret` encoding transcribes position information relative to an estimated level of the hand on the fretboard. This encoding aims at facilitating the identification of translated positions, and encouraging the learning of musical knowledge up to transposition. It includes 25 binary values to encode the level of the hand and $6 \times (s + 1)$ values to encode for each of the 6 strings, 1 open string position and s relative positions that are assumed to be accessible without shifting the level of the hand on the fretboard. The parameter s is called the *HandSpan* and is commonly given the value 5 given that a normal hand is unlikely to cover more than this number of adjacent frets on a standard guitar.

The `StringFret` and `RelativeStingFret` encodings possibly include six additional values to specify if a sounding string is attacked or held from the past. Hold values have indeed shown to play an essential role in the modeling of music with neural networks (Hadjeres, Pachet, and Nielsen, 2017).

On tablature token encodings As it will be further detailed in Chapter 4, sequence to sequence neural networks such as the transformer have gained an important popularity in symbolic MIR in the last years. These models generally work with a representation of music as sequences of atomic values called *tokens*, similarly to sequence of words in text. Figure 3.4 illustrates a token representation of guitar tablature data proposed by Chen et al. (2020). Technically, tokens are fed to neural networks as one-hot vectors, meaning vectors having one unique value at 1 and the others at 0. One-hot representations are associated with a dictionary and allow the encoding of sets of possibly unrelated elements, which contrasts with the previous encodings. As illustrated on Figure 3.4, encoding a given fretboard position will typically require several consecutive tokens, including one for the string and one for the fret. This method consequently produces longer sequences than the previous many-hot encodings, which can encode in one unique vector a whole instantaneous position on the guitar fretboard. Sarmento et al. (2021) have also proposed a tablature dedicated token representation, which can additionally encode other type of instrumental tracks including drums and bass.

3.3.3 *MySongBook* statistics

The parsing and encoding tools described in the previous section facilitate the computation of corpus statistics. Figure 3.5 displays the proportions of fretboard positions within the whole MySongBook corpus, highlighting the predominant use of the lowest frets (73% on open strings or on fret ≤ 5). Figure 3.6 illustrates the most common chord positions, using at least four strings, of the corpus. The top line

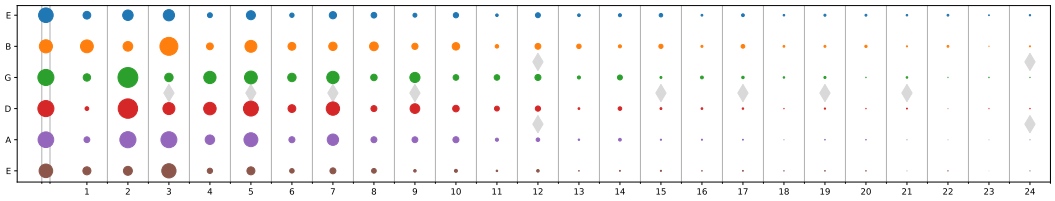


FIGURE 3.5: Occurrence of string/fret combinations across the whole MySongBook corpus (Cournut et al., 2021).

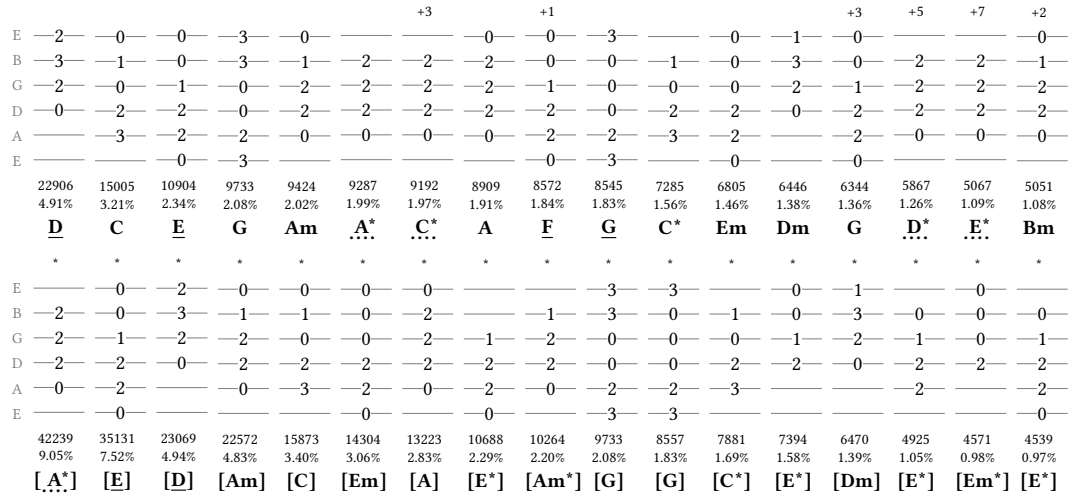


FIGURE 3.6: Most frequent *instantaneous* positions implying at least four strings, encoded with RelativeStringFret (top, absolute positions) and RelativeStringFret* (bottom, relative positions, with omission of the RootFret value), with their number of occurrences and the ratio of their occurrences in the corpus. Positions notated with a star, like A*, are "sub-chords", meaning that at least one string could be added (generally the top string) to get another usual position (Cournut et al., 2021).

shows *exact* positions while the bottom line shows *relative* positions, possibly performed at any level of the fretboard using barre chords. These statistics enable to compare how the different positions tend to be translated across the corpus. They also show the surprising prominence of some "sub-chords", illustrating some aspects of guitar practice in this repertoire (Cournut et al., 2021).

Although these statistics provide global insights on western guitar practice, they nevertheless require to be computed on separated sub-corpora for the study of distinct musical styles. Figure 3.7 compares the composition of chords in different styles. The figure highlights for instance the predominance of fifth chords, also called *power chords*, in the metal style as well as the rarity of these chords in the classical style ("5" on the top line of the figure).

These statistics are further discussed and interpreted in our article (Cournut et al., 2021).

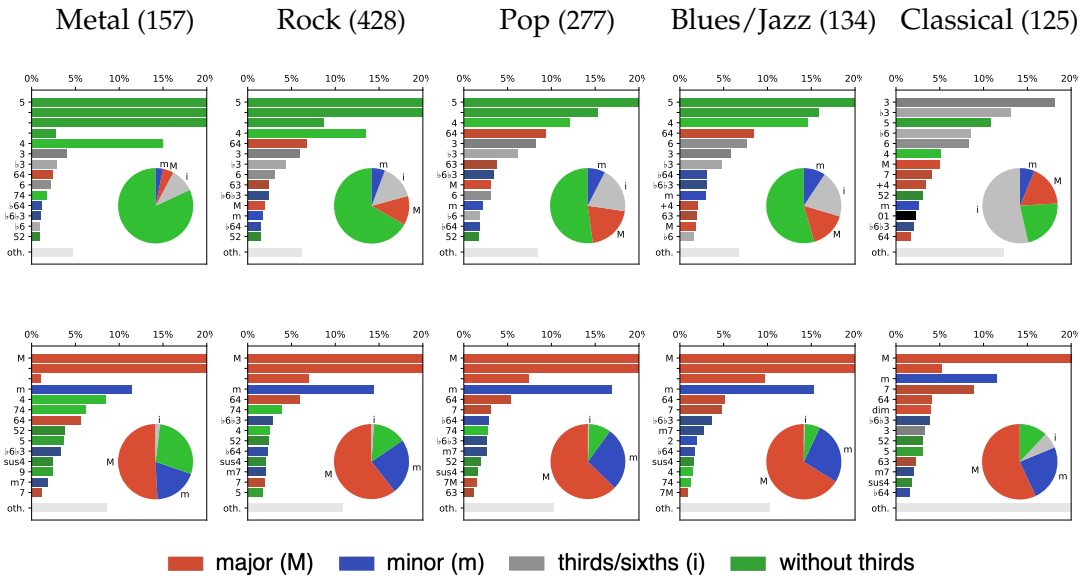


FIGURE 3.7: Frequencies of chord types in several genres, considering chords with 2 or 3 strings (top), or with 4, 5, or 6 strings (bottom), and gathered into families: major, minor, thirds and sixths, chords without thirds. After each genre is displayed the number of pieces in that genre. To allow to compare more precisely less frequent chords, data over 20% concerning the most frequent chord are split on several lines (Cournut et al., 2021).

3.3.4 Playing function prediction

As mentioned in Section 3.1.2, the guitar instrument can play considerably different musical functions, especially within large instrumental formations in which the different musical layers, such as melody, harmony, bass, rhythm, are commonly split among the instruments. Most common functions are rhythm guitar and lead guitar although a substantial part of the pop/rock repertoire can not unambiguously be classified within one of these two categories as illustrated in Figures 3.2c and 3.2d.

Predicting playing function in tablatures potentially allows the identification of consistent sub-datasets limited to one specific musical function. While a general tablature generation model would typically benefit from being trained on a mix dataset in order to potentially generate tablature with any kind of function, a model intended to only assist a specific sub-task of the composition process would in turn benefit from being trained only on the type of data involved in this task. For instance, it is probably not desirable to train a model dedicated to rhythm guitar composition on lead guitar data and vice-versa. Predicting tablature playing function is therefore considered as a key task to elaborate machine-learning tools to assist punctual composition situations. The rest of this section presents a method for the specific task of rhythm guitar detection, that was elaborated by D. Régnier during its Master internship that I supervised, and which reached to a publication (Régnier, Martin, and Bigo, 2021).

Following the methodology proposed in Section 1.3, a set of 31 high-level textural features computed at the bar level have been identified as being potentially correlated with the musical function played by the guitar. As investigated in Section 2.3.2

note features		chord features		tab features	
# notes	$7 \cdot 10^2$	chords*	$2 \cdot 10^3$	min fret	$2 \cdot 10^3$
single notes*	$1 \cdot 10^3$	# 2-chords	$1 \cdot 10^1$	max fret	$2 \cdot 10^3$
min pitch	$3 \cdot 10^3$	# 3-chords	$3 \cdot 10^2$	mean fret	$2 \cdot 10^3$
max pitch	$8 \cdot 10^2$	# 4-chords	$5 \cdot 10^2$	min string	$3 \cdot 10^3$
mean pitch	$2 \cdot 10^3$	# 5-chords	$2 \cdot 10^2$	max string	$4 \cdot 10^0$
pitch ambitus	$1 \cdot 10^3$	# 6-chords	$9 \cdot 10^1$	mean string	$7 \cdot 10^2$
pitch variety	$2 \cdot 10^3$	chord variety	$9 \cdot 10^2$	*l-r(s)**	$1 \cdot 10^2$
min interval	$3 \cdot 10^1$	m/M triad*	$5 \cdot 10^2$	*l-r* (100%)*	$1 \cdot 10^2$
max interval	$1 \cdot 10^{-1}$	fifth interval*	$1 \cdot 10^2$	*w.b(s)**	$6 \cdot 10^0$
interval var	$2 \cdot 10^2$			*bend(s)**	$2 \cdot 10^3$
duration var	$1 \cdot 10^2$			*l-h vibr(s)**	$8 \cdot 10^2$

TABLE 3.1: Features describing tablature bars for the rhythm guitar detection task. Binary features are indicated with a *. The importance of each feature in the dataset is indicated by its ANOVA *F-value* (Régnier, Martin, and Bigo, 2021).

with the classical piano repertoire, the texture of a score region results from several musical features including rhythmic patterns, note density, diversity and register. In the case of string instruments, these textural features are also linked to the choice of strings that are played as well as the level of the hand on the instrument fretboard.

The selected features encode information regarding note pitches, onsets, durations, string and fret indications, as well as annotations of some technical playing techniques including *Let-Ring* (*l-r(s)*), *whammy bar* (*w.b(s)*), *bend(s)* and left-hand vibratos (*l-h vibr(s)*). Note that some features may derive from combinations of others. For example, the pitch of a note can be deduced from its string and fret value, as we only considered tablatures for guitars with standard tuning. Table 3.1 indicates the whole set of features.

During its internship, D. Régnier used its musical expertise to manually annotate 102 guitar tablatures from the MySongBook corpus, specifying at each bar if it was rhythm guitar or not. The resulting dataset included 6051 rhythm guitar bars (82% of the whole set of annotated bars). Different functions were identified within the complementary class including solos, licks, riffs and studio arrangements. Annotations and computed features for all bars of the dataset were publicly released[7].

Model explainability Figure 3.8 shows the value distribution of a selection of features extracted from bars of both classes in the annotated dataset. To facilitate the comparison of the two classes, the histograms indicate the proportion of feature values in each class rather than the actual number of bars. As expected, rhythm guitar and non-rhythm guitar bars appear to be respectively correlated with the presence of chords and the presence of single notes . Non-rhythm guitar bars can also be distinguished by a lower number of notes and distinct chords. Rhythm guitar bars can finally be distinguished by a lower register that appears in pitch, fret, and string related features. An ANOVA Fischer test is performed for each feature as an indication of its correlation with the two classes. The results are displayed in Table 3.1.

[7] https://gitlab.com/lbigo/rhythm-guitar-detection

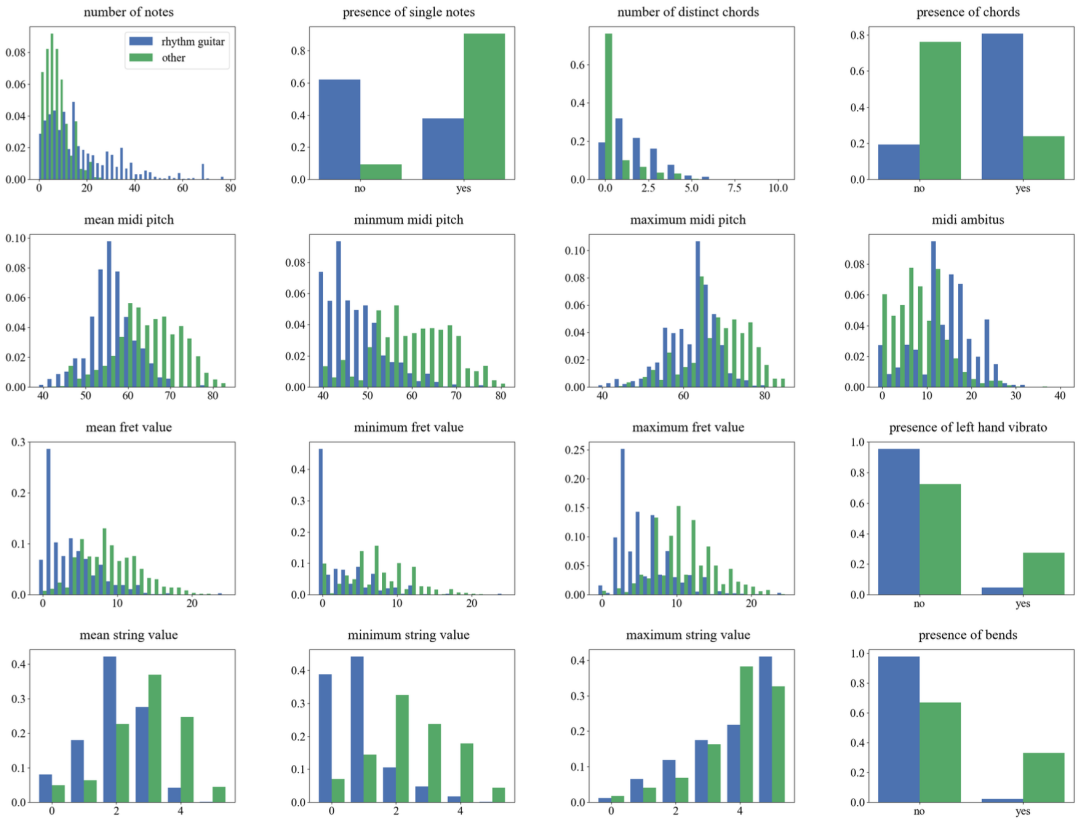


FIGURE 3.8: Distribution of some features on bars annotated with labels *Rhythm guitar* (dark, blue) and *other* (light, green) (Régnier, Martin, and Bigo, 2021).

Evaluation Following the *leave-one-piece-out* validation process described in Section 1.3, an LSTM neural network is trained to predict if a tablature bar should be labelled as rhythm guitar or not given its set of computed features. A recurrent model was selected for this task in order to model the tendency of adjacent bars to have the same function[8]. Different models, resulting from various combinations of features, were evaluated and compared as shown in Table 3.2. We consider a baseline model that only looks at the presence of chords and single notes in each bar. We then evaluate score based features (first two columns of Table 3.1) and tablature based features only (third column of Table 3.1). Finally, we evaluate a model taking into account the whole set of features. In addition to F_1 score, Table 3.2 displays the *precision* and the *recall* on rhythm guitar label predictions, which might distinctively be used as evaluation metrics depending on the foreseen model application.

On the one hand, maximizing *precision* penalizes false positives and potentially leads to the creation of a consistent rhythm guitar sub-corpus although possibly small and uniform. Such a corpus would facilitate the training of a model that is expected to produce *typical*, but not necessary surprising, rhythm guitar tablatures. On the other hand, maximizing *recall* penalizes false negatives and potentially leads to a larger sub-corpus with more diversity although more sparse and including more debatable rhythm guitar examples. Such a corpus would be appropriate for the training of a model which is intended to be creative, outputting rhythm guitar tablatures

[8] a number of tablatures are actually labelled by one unique function during the whole piece

	r.g precision	*r.g* recall	**F_1 score**
chords/single notes presence	0.86	0.88	0.87
note + chord features	0.95	0.94	0.94
tab features	0.95	0.93	0.94
all features	0.96	0.94	0.95

TABLE 3.2: Precision, recall and F_1 score obtained for the prediction of rhythm guitar (*r.g*) with an LSTM trained on different combinations of features.
(Régnier, Martin, and Bigo, 2021)

that possibly diverge from the common definition of rhythm guitar. In complement, it should be noted that for a classifier that outputs a probability, like neural networks do, moving the decision threshold, which is generally set by default to 0.5, could also be a way to balance between consistency and variety.

As displayed in Table 3.2, the model trained on the whole set of features reaches an F_1 score of 0.95, that is 8% better than the baseline model, which only takes into account presence of chords and single notes. To get a better estimation of the relevance of the different categories of features, two additional models were evaluated, the first one without any tablature features, the second one with tablature features only. Interestingly, these two last models, although trained on two disjoint sets of features, both compete with the best model.

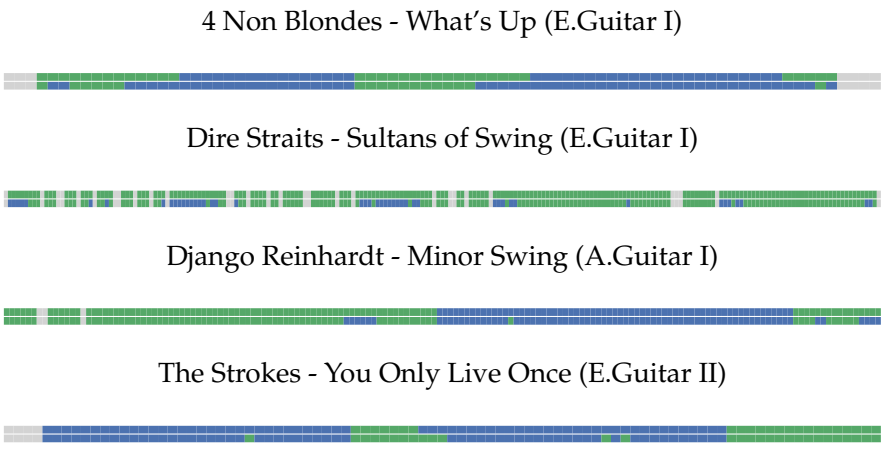


FIGURE 3.9: Comparison of manual annotations (top lines) and predictions (bottom lines) of a some tablatures of the dataset. Sections labelled as rhythm guitar are displayed in blue. Other sections are displayed in green. Empty bars are left in gray.
(Régnier, Martin, and Bigo, 2021)

Figure 3.9 displays a comparison between reference annotations (top line) and predictions (bottom line), with the "all-features" model, for some tablatures of the annotated corpus. Although the model succeeds in identifying large scale sections, it still predicts unlikely short sections, sometimes for one unique bar. Comparing statistics on predictions and reference annotations highlights the difficulty of the model to predict continuous rhythm guitar sections. In particular, the model tends

to detect isolated rhythm guitar bars whereas the reference annotation do not include any of them. Surprisingly, the use of a bidirectional LSTM did not significantly reduce the prediction of isolated rhythm guitar bars. The limited amount of training data might explain the difficulty of the network to efficiently learn switch tendency between the two labels. Future experiments include testing sequence models, such as Hidden Markov Model, that might better manage limited training data.

Figures 3.10a and 3.10b illustrate two examples of false negatives, *i.e.* rhythm guitar bars predicted as being non-rhythm guitar bars (the 2nd bar in both examples). In the first example, the wrong prediction occurs at the final bar of a musical phrase, which leads to a new phrase beginning on the next bar. The rhythm guitar punctually plays a short melodic lick often referred as a *fill*, similarly to the example in Figure 3.2d. In the example 3.10b, the guitar starts to play bass single notes and produces a melodic line which is wrongly estimated by the model as non-rhythm guitar. This behavior could arguably be qualified as being at the edge of the common definition of rhythm guitar when looking at the guitar tablature isolated. One natural track of improvement of this method is therefore to take into account the other tracks of the song, especially the singing part.

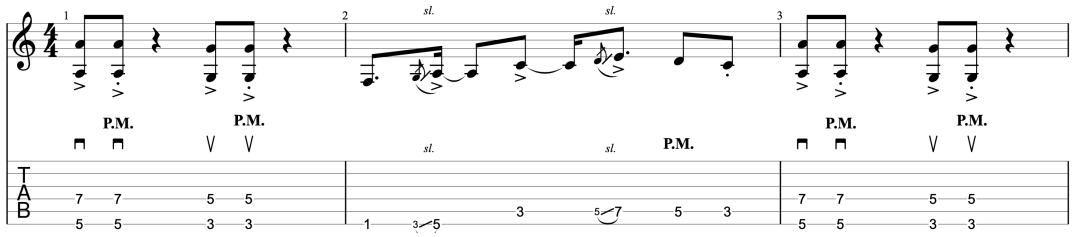
Figures 3.10c and 3.10d illustrate examples of false positives, *i.e.* non-rhythm guitar bars predicted as rhythm guitar bars. The first example includes an extract of a solo part where the guitar repetitively plays arpeggios of the underlying chord sequence. In spite of the high register, which is unlikely for rhythm guitar sections, the model is probably misled by the repetition and low variety of the tablature content, as well as the presence of perfect triads, these features being predominantly correlated with rhythm guitar sections. The example 3.10d is extracted from a jazz solo. The model is clearly misled by the sudden occurrence of chords here. As it is often the case in jazz solos, the melody punctually turns into successive chords which do not necessarily feet the underlying chord sequence. This behavior commonly lasts a few bars before going back to a monophonic melody.

When interpreting these results, it is worth remembering that the small amount of wrong predictions is likely to correspond to *limit cases* of rhythm guitar whose inclusion/exclusion in a rhythm guitar dataset might not be crucial due to the relative subjectivity of what rhythm guitar is.

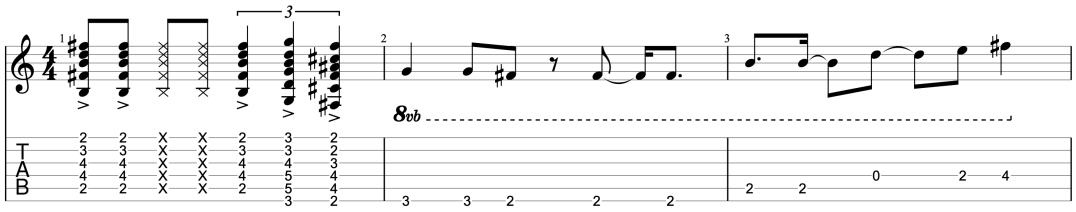
3.3.5 Rhythm guitar texture imitation

An accompaniment part in modern popular music is generally driven by an underlying sequence of chord symbols, which is rendered with a particular texture contributing to the style of the piece. Figure 3.11 illustrates a singing part accompanied by a rhythm guitar part and a bass guitar part, following the chord sequence [F, Dm]. Rhythm guitar parts, as most accompaniment parts in this style, commonly render successive chord symbols with a similar texture in order to conserve a style uniformity over the song. Figure 3.12 displays two score examples of rhythm guitar region illustrating this texture uniformity.

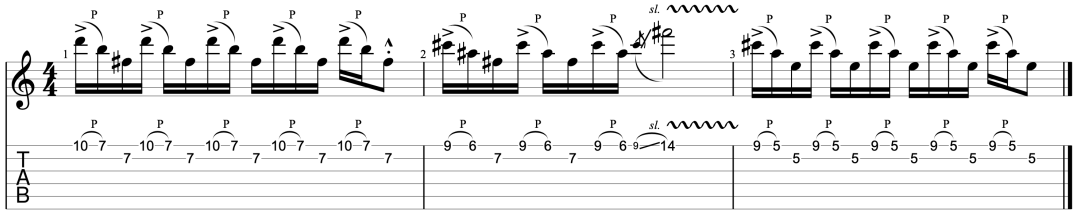
This section describes an experiment for the task of *rhythm guitar continuation by texture imitation* which can be illustrated by Figure 3.13. A model predicts the tablature of a chord region (in green on the figure) given its labelled chord symbol (G on the figure) and the tablature of the previous chord region (in blue on the



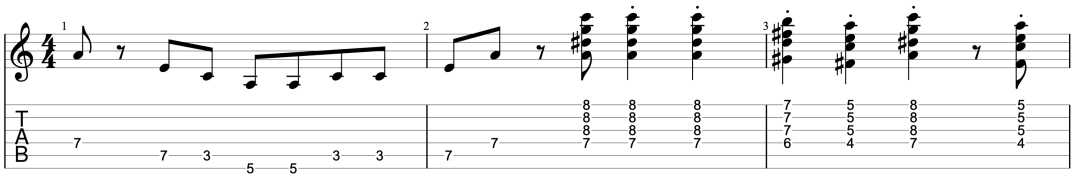
(A) *Stairway To Heaven* (Led Zeppelin)



(B) *When The Sun Goes Down* (Artic Monkeys)



(C) *Hotel California* (Eagles)



(D) *Minor Swing* (Django Reinhardt)

FIGURE 3.10: Examples of false negatives (A and B): the second bar is wrongly predicted as non-rhythm guitar on both extracts. Examples of false positives (C and D): the second and third bars are wrongly predicted as rhythm guitar on both extracts. Tablatures rendered with Guitar Pro.

figure), assuming the continuity of the texture. This task is addressed as a supervised machine learning problem, where a model is trained with a set of couples of adjacent rhythm guitar chord regions which are assumed to feature a similar texture, like in the examples in Figure 3.12. The training set is limited to rhythm guitar sections of MySongBook which are obtained thanks to the method presented in Section 3.3.4. Only rhythmically similar couples were then selected. A Gated Recurrent Unit (GRU) neural network with two hidden layers was trained to predict the second region given its labelling chord symbol and the previous region. The regions were encoded with the *StringFretWithHold* encoding detailed in Section 3.3.2. Chord symbols were represented by pitch-class vectors.

Evaluation Figure 3.14 illustrates various outputs of the model. Each time, the content of the bar A accompanied by the chord label of the bar B_r are given as input

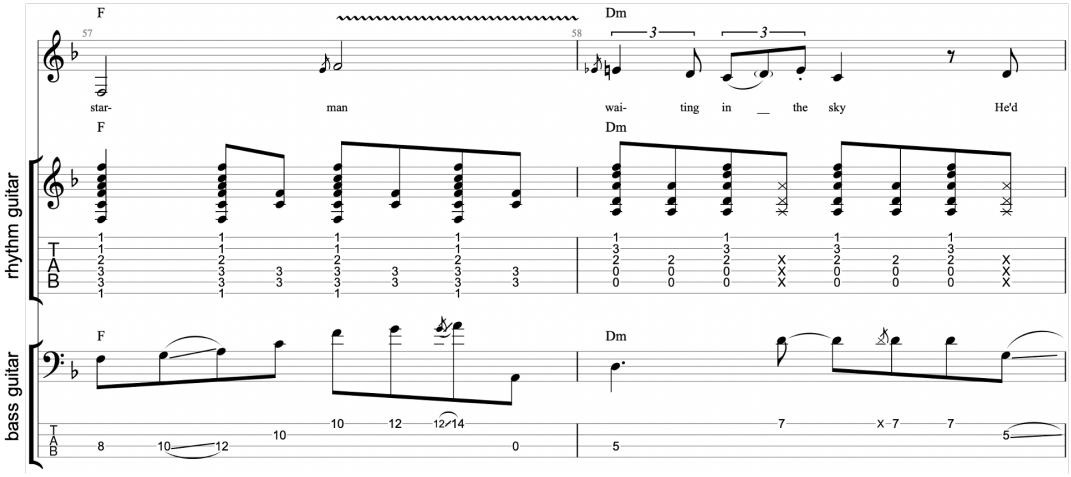
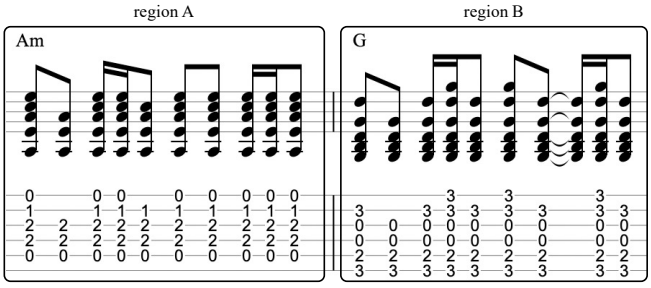


FIGURE 3.11: A lead melody (top) accompanied by a rhythm guitar part (middle) and a bass part (bottom), extracted from the song *Starman* (David Bowie).

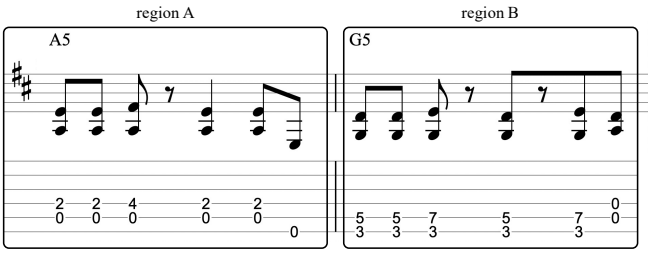
to the model, which outputs the prediction B_p which we qualitatively compare to B_r. The example in Figure 3.14a illustrates an exact prediction, which is here facilitated by a constant rhythm over the bars, as well as the use of two very common chord positions (chords *D* and *A*), as confirmed by the statistics on corpus chord positions detailed in Figure 3.6. Figure 3.14b illustrates a wrong prediction, partially due to an unexpected change of rhythm between bars A and B_r. In addition to the rhythm change, the bar B_r includes an unprecedented use of an open string, indicated by fret 0, while the model chooses the fret 7 at the second string, therefore inducing an inversion on the chord (Am/E instead of Am) in B_p. Figure 3.14c illustrates the difficulty of the model to generate single notes, which almost systematic fails on arpeggio texture imitation, and could presumably be improved by taking into account *Let Ring* playing techniques that are almost systematically employed in arpeggios. Finally, Figure 3.14d illustrates the generation of an unlikely strumming B_p which contrasts with the regularity of the initial bar (A) and the reference bar (B_r) and illustrates some limits of the current model to produce playable tablatures.

Evaluating the output of a generative model is known to rise a number of questions as it generally relies on the user/listener subjectivity. Taking advantage of the supervised training framework of our continuation task, we focus our evaluation on the ability of the model to output a tablature content similar to the expected one. For this purpose, we use three high-level descriptions of tablature region, which respectively focus on rhythm, pitch-class and fretboard position content. Given a tablature region X, we notate $\mathrm{o}(X)$ the set of sixteenth positions at which one or more notes have their onset. We notate $\mathrm{pc}(X)$ the set of pitch-classes of the notes included in the regions. We finally notate $\mathrm{p}(X)$ the set of string/fret combinations of the notes included in the region. The region illustrated on the top left of Figure 3.13 features the following descriptions: $\mathrm{o}(A) = \{0, 2, 4, 5, 6, 8, 10, 12, 13, 14\}$, $\mathrm{pc}(A) = \{0, 4, 9\}$ and $\mathrm{p}(A) = \{(1, 0), (2, 1), (3, 2), (4, 2), (5, 0)\}$.

The results displayed in Table 3.3 indicate that our model, when trained on rhythmically similar couples ($\mathrm{o}(A) = \mathrm{o}(B)$), reaches a F_1 score of 0.67. 35% of the vectors are correctly predicted and 5% of second regions are perfectly predicted (14% if we



(A) Rhythm guitar region from *Wish You Were Here* (Pink Floyd)



(B) Rhythm guitar region from *There's Gonna Be Some Rockin'* (AC/DC)

FIGURE 3.12: Two rhythm guitar tablature extracts illustrating texture uniformity across successive chord regions. Textural features similar from one bar to the next one include rhythm, ambitus, note density and string/fret positions.

exclusively evaluate rhythmically similar couples). A large part of the inexact predictions of B succeed however in perfectly replicating the score content in terms of onsets, pitch-classes, and to a lesser extent, positions as shown by the third, fourth and fifth columns of Table 3.3. Although these prediction rates seem low, we remind that one exact B region prediction consists in $157 \times 16 = 2512$ binary values[9] that must all be correctly predicted which makes the problem relatively hard if we consider the unpredictable nature of music despite its frequent uniform texture in accompaniment parts. Additionally, it must be reminded that if the prediction of B is used as a (strict) way to evaluate our model, the ultimate goal is to generate tablature regions that would be plausible enough to be used in a context of assisted composition, which is arguably a less demanding requirement than the ability to predict an upcoming region within a musical piece. Using a training set limited to rhythmically similar couples obviously limits the ability of the model to learn rhythmic variations that potentially occur over adjacent regions, but it nevertheless provides a more consistent model, which is more likely to output acceptable content although with limited originality. As for the playing function prediction task detailed in Section 3.3.4, evaluating this trade-off should be done according to the expected use of the method in a composition process. On the one hand, a consistent model might be useful to assist the functional composition of simple accompaniment parts, providing a basis allowing to focus on another layer, for instance in a

[9] 157 values resulting from the *StringFretWithHold* encoding, and 16 time steps in the common case of a 4 beats measure divided in height [eighth] note slices.

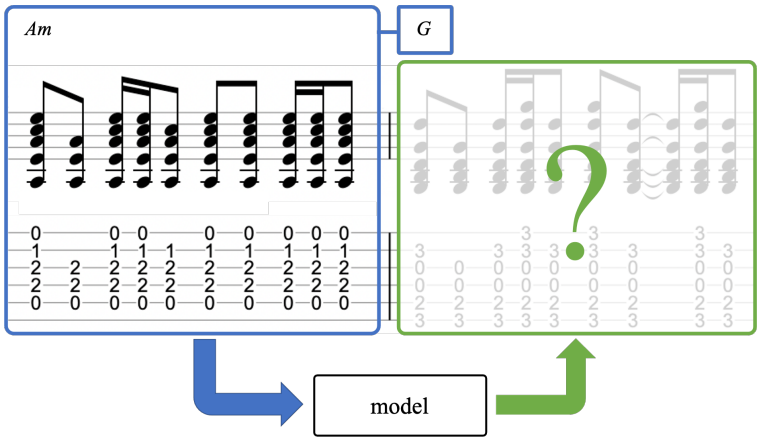


FIGURE 3.13: The task of rhythm guitar continuation by texture imitation applied on two adjacent bars. A model predicts the tablature content of the second bar given its chord label and the content of the previous bar, assuming the continuity of the texture.

F_1	$B_p[i] = B_r[i]$	$p(B_p) = p(B_r)$	$o(B_p) = o(B_r)$	$pc(B_p) = pc(B_r)$	$B_p = B_r$
0.67 (0.8)	35% (46%)	18% (24%)	25% (66%)	46% (51%)	5% (14%)

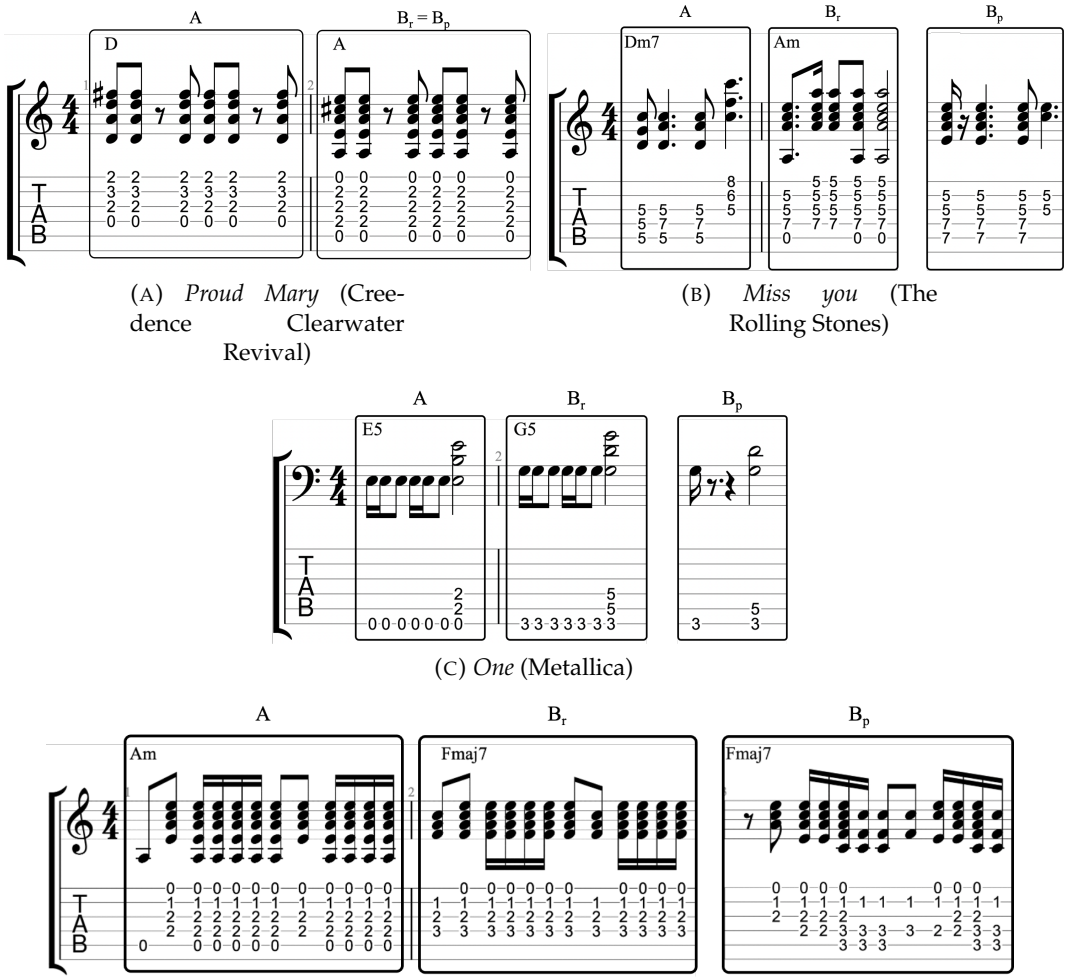
TABLE 3.3: Evaluation of our model on the task of predicting the region B from its chord label and the region A. The model was trained exclusively on couples (A,B) with o(A) = o(B). The numbers in brackets (x) are obtained by evaluating the models exclusively on couples that are rhythmically similar (o(A) = o(B)), considering the others as highly difficult to predict.

pedagogical context. On the other hand, a more versatile model trained on possibly dissimilar couples could lead to more unexpected outputs, stimulating the creativity of the composer but with a presumable need of post-generation corrections and re-writing. In particular, the ability of a model to allow texture variations would seem particularly promising for the composition of contrasting chord regions that typically occur in this repertoire after three uniform occurrences as formalized by the System and Contrast Model (Bimbot et al., 2016).

3.4 Perspectives

The continuation of this research on guitar tablature modeling is described in details in the proposal of the 4-years TABASCO ANR JCJC project (TAblature ASsisted COmposition) that I am coordinating from October 2022. This perspective section aims at giving a synthetic overview of these upcoming works. As I will dedicate most of my research time of the next years on this project, this perspective section is more detailed than the ones of Chapter 2 and 4.

After a brief description of the goals of the project, we will introduce its three main parts which respectively focus on composition practice studies, experimental algorithms and software contributions.



(A) *Proud Mary* (Creedence Clearwater Revival)

(B) *Miss you* (The Rolling Stones)

(C) *One* (Metallica)

(D) *One* (Johnny Cash, original song: U2)

FIGURE 3.14: Prediction of the second bar (B_{pred}) of different contiguous couples $A + B_{\text{ref}}$ of the dataset.

3.4.1 Objectives

The project aims at elaborating algorithmic methods to assist musicians, of all skills and musical profile, in the task of composing and writing guitar tablatures in musical styles that can be grouped under the general term of modern popular music. These methods notably include machine-learning approaches benefiting from corpus data. They do not aim at composing music but rather at proposing tools to the composer, on selected situations that commonly arise during the composition and writing of guitar parts. In this sense, these situations might be more easily assimilated to music *arrangement* than music *composition*. This includes for instance the continuation of an accompaniment part, the transfer of the style of a tablature extract into another one, or the fine-tuning of style expressiveness in a tablature. While the composition of the score's heart components, such as chords and melody, is left to the composer, the TABASCO tools aim at facilitating the rendering of these elements through a concrete tablature, with a fine control of stylistic and expressiveness aspects.

The use of the proposed methods can be beneficial in two ways. On the one hand, they aim at assisting composers with limited guitar skills in creating realistic guitar parts, for example to accompany a lead singing part or other instruments in a band, therefore opening guitar music composition to a wider public including beginner guitarists and even non-guitarists. On the other hand, they aim at stimulating the creativity of experimented guitarists by encouraging them to experiment musical choices outside their compositional habits. Essential for this second objective, this project aims at elaborating tools with substantial control possibilities, thus maintaining the composer at the heart of the creative process. In addition to the creation of composition tools, the project aims at improving our knowledge on modern popular music composition, which remains a relatively unknown field due to the variety of practices.

From the musicology point of view, it is assumed that the study of modern popular music practices in the guitarist community, by means of corpus study and composer practice surveys, will enable us to precisely specify innovative algorithms that can meet the composer's needs in common composition scenarios, in particular relating to style expression. For the reasons detailed in Section 3.1.3, this project contributes to a major challenge in modern popular music, which is to motivate, by the use of innovative computational tools, the reestablishment of the musical score as a key tool for the composition of popular music. In order to maximize the impact of these methods on the composer community, some of them will be selected and distributed as plugins embedded in the open-source reference music notation software MuseScore, which is used by a circle of musicians much larger than the guitarist community.

From the computer science point of view, the project is built on the hypothesis that the elaboration of such tools has become possible thanks to the recent advances of machine learning algorithms, as well as the substantial quantity, quality and variety of guitar tablatures that are necessary to efficiently train these algorithms and which are made available thanks to our collaboration with Arobas Music (the MySongBook corpus) as well as some recent open corpora such as DadaGP (Sarmento et al., 2021).

Although these algorithms will be trained at presuming some information that is hardly predictable because of the intrinsic diversity of music, it is assumed that this training procedure ultimately enables the model to produce plausible, and diverse, outputs that can arouse the interest of the composer. Furthermore, limiting the intervention of algorithms to well-defined subtasks of the composition process also contributes to address this challenge of reconciling the divergence between the open-ended and under-determined concepts of artistic creation and the machine-learning paradigm that relies on well-defined problems and quantifiable performance (Gioti, 2021).

3.4.2 Composition practice studies

We first plan to conduct a composition practice survey focusing on guitar music in the modern popular style. This study will aim at studying the impact of music notation software on the composition process, and identifying composition and notation contexts in which the composer's experience could be enhanced by algorithmic

tools, especially focusing on accompaniment part composition and gesture annotations. In addition to identifying the functionality of these tools, the survey will aim at specifying the user interface and the control possibilities that the tools should provide when they will be implemented as music composition software plugins. In a second survey, coming in the end of the project, composers will be asked to experiment and evaluate the tools produced by the project. The two surveys will involve, as much as possible, the same pool of participants. In addition to the specification of future algorithms, we believe that these survey will generally contribute to delimit and improve our knowledge in modern popular guitar music composition practices.

3.4.3 Tablature arrangement algorithms

The main part of the project will consist in elaborating machine-learning algorithms intended to assist tablature arrangement. These research axes will be at the center of the PhD of A. D'Hooge. They can be divided into two categories being accompaniment parts composition and gesture annotation modeling.

Accompaniment parts composition

Accompaniment composition algorithms will meet two specific composition tasks being the composition of a rhythm guitar section given a chord sequence and a reference texture and the composition of a bass guitar section given a rhythm guitar track tablature. In addition, it is planed to investigate sophisticated methods based on the use of variational autoencoders allowing an intuitive navigation within generated outputs for these tasks. In a general way, a particular attention will be given regarding the ability of the algorithms to be controlled by the user, according to increasing concerns in the MIR field on this question (Briot and Pachet, 2020).

Figure 3.11 illustrates a score excerpt of the song *Starman* (David Bowie). The extract includes two successive chord regions (F and Dm) with both a rhythm guitar track and a bass guitar track.

Rhythm guitar part composition This axis is in direct continuation with the work presented in Section 3.3.5. The perspectives of this work include an extension of the generation process to render a generic array of chord symbols given a reference texture, including the rendering of chord regions with variable length. Taking into account the underlying metric of the chord symbol sequence could additionally allow to vary the versatility of the model depending on the relative positions of the chord regions, therefore encouraging diversity at theoretical contrasting regions as proposed by Bimbot et al. (2016). The architecture of the model could also be improved for instance by inputting the conditioning chord symbol between an encoding and a decoding stack of layers, as it is commonly done in encoder-decoder architectures. In such architecture, the encoding part is assumed to extract the texture of the input region, while the decoding part is assumed to render this texture given an arbitrary chord symbol (Wang et al., 2020), possibly helped by the use of an adversarial mechanism (Kawai, Esling, and Harada, 2020). Beyond the continuation of a tablature extract, this method is felt to be usable for style transfer tasks where the idea is to interpret a song A in the style of another song B, which has raised a number of MIR researches lately (Dai, Zhang, and Xia, 2018; Cífka, Şimşekli, and Richard, 2020).

Bass guitar track composition Similarly to rhythm guitar, the musical role of the bass guitar part strongly relates to the realization of the song's underlying chord progression. We plan to elaborate a model able to generate bass tablature suggestions intended to accompany an input rhythm guitar tablature region. Such a tool primarily aims at assisting non-bassists wishing to add a bass track to accompany a guitar part being composed. Additionally, making the model controllable will also promote its use by experimented bass guitar composers aiming at diversifying their compositional habits, following research initiatives in the audio domain (Grachten, Lattner, and Deruty, 2020). To address this goal, we will experiment an approach that consists in training a sequence-to-sequence neural network, LSTM or transformer, to predict the content of a bass track tablature given an aligned rhythm guitar tablature. The MySongBook corpus provides such alignments for most included songs. As this task can be reformulated as a sequence translation problem, a particular look will be given to recent research successfully adapting Neural Machine Translation methods to the musical domain (Makris, Agres, and Herremans, 2021). The experiments will be done with various stylistic homogeneity of the training corpus. The model will be made controllable by the addition of target textural features (density, diversity, and compatibility with a style label) specified along with the input rhythm guitar tablature.

Navigating into accompaniment spaces Going further into assisted accompaniment composition approaches, a perspective of this research is to investigate the unsupervised learning of accompaniment texture spaces, in which the composer could easily navigate and select suggestions during its composition process. These navigation spaces will aim at suggesting accompaniment textures made of rhythm guitar, bass guitar, or both. The textures will be organized in these spaces by proximity, presumably reflecting style relations. These spaces will be built by training variational auto-encoders (VAEs) (Kingma and Welling, 2013) on tablature chord regions. During their training, VAEs will be building latent spaces aiming at organizing the datasets training textures with musically meaningful distances. Posterior to the training, the models will enable the sampling of unseen textures by interpolation in their latent spaces. VAEs have proven to be promising in human/machine music co-creativity (Wang et al., 2020; Kawai, Esling, and Harada, 2020; Grachten, Lattner, and Deruty, 2020; Roberts et al., 2018; Esling et al., 2019) but have not been investigated yet in our knowledge in the specific case of symbolic guitar tablatures. A major challenge in this task will be the musical interpretation of the dimensions of the learned latent spaces and their controllability in the context of user plugins.

Unsupervised approaches for texture modeling are an important part of our research perspectives for both guitar music and piano music as described in Section 2.5. We plan to investigate this axis on both repertoires in the two next years in the context of the PhDs of A. D'Hooge and L. Couturier in close collaboration.

Gesture annotation modeling

The inclusion of fingering and playing techniques annotations in the MySongBook corpus enables to address their prediction as supervised machine-learning tasks. These prediction methods will be implemented as tools aiming at (1) facilitating the

writing/transcription of non-guitar music into playable and expressive guitar tablatures and (2) experimenting different rates of expressiveness in a tablature region. Importantly, the methods will let the user have fine control over the prediction in order to adapt the output to a targeted style and level of playability.

Position estimation As mentioned in Section 3.1.1, the position choice, also called *fingering*, is an inherent task to the guitar and other string instruments enabling the playing of a same pitch at different positions of the instrument. Tablatures specify string/fret combinations for each note in order to dispense the performer from this decision. Transcribing a non-guitar musical piece into a guitar tablature requires performing these position choices, which is generally a non-trivial task especially for beginners. While most of the studies so far have proposed to compute an optimal sequence of positions based on the musical score content, practice studies have shown that position choices also depend on out-of-score context such as the proficiency of the player (Yazawa, Itoyama, and Okuno, 2014) as well as its playing style (McVicar, Fukayama, and Goto, 2015). This task will pursue two objectives. First, it aspires to improve the state of the art in automatic position estimation based on in-score information. Secondly, it aims at making such models controllable in order to let the user navigate between different optimal position solutions that differ in terms of style and level of playability. The first objective will be addressed by taking advantage of cutting-edge deep learning techniques, in particular attention-based neural networks, as well as unprecedented large and high-quality quality training sets including the MySongBook corpus and the DadaGP corpus (Sarmento et al., 2021). The second objective will be addressed by informing the model at the training stage about indications regarding the level and the musical style of the tablature, which are available through text tags in the MySongBook corpus. At the prediction stage, the model will then be usable with variable constraints on style and level.

Playing technique prediction This task aims at elaborating models for the prediction of playing technique annotations, as presented in Section 3.1.1, which has never been addressed in symbolic tablatures to our knowledge. Such methods aim at being used by non-expert guitarists to fine-tune the expressiveness of a tablature being either composed or transcribed from a non-guitar score. Playing technique prediction models will be experimented on both low level and high-level tablature representations. High level representations will relate to melody, harmony, rhythm or texture, that will be found to be correlated with playing techniques annotations following some preliminary results obtained by Q. Normand during its Master internship that I supervised (Normand, 2021). In a first experiment, the automatic computation of these features will let us approach the playing technique prediction task as a supervised classification problem where the occurrence of each technique is independently estimated on each element (notes or chords) of the tablature given its context. In a second experiment, sequence models such as LSTM and Transformers will be used to model the dependency between nearby playing techniques annotations and make the prediction on all elements of a region in a row. In order to make this model controllable by the user on different axes, such as expressiveness and musical style, we will take advantage of the property of neural networks to output prediction probabilities, letting the user gradually move the decision threshold to experiment various amounts of expressiveness. Regarding style, the user will

have the possibility to select a model trained on different style-specific sub-dataset, which is made possible thanks to style-related labels available in the MySongBook corpus. The interpretation of playing technique statistics as well as their relation to musical style will be studied in close collaboration with the musicologist B. Navarret given his past research related to guitar playing techniques and composition practices in the specific case of contemporary music composition (Lähdeoja et al., 2010). A particular attention will be given to the occurrences of *bended* notes within typical pentatonic patterns and their ability to contribute to the establishment of major modern popular guitar musical styles.

3.4.4 Software contributions

We finally plan to contribute to several open-source music projects used by the MIR research community, Music21, Dezrann and MuseScore, to improve their ability to process guitar tablatures. These contributions will serve the needs of the algorithms described above, but they will also aim at facilitating and encouraging research and pedagogy initiatives of the academic community around guitar music.

Contributions to Music21

Despite substantial efforts in the last years, the support of tablatures in the python library music21 is still sparse and unstable. We plan to gather a set of pull requests throughout the project to improve guitar related features including the processing of playing techniques, tuning/instrument information that could improve the visual/audio rendering within music notation software such as MuseScore or Guitar Pro, as well as the accurate reading and writing of tablatures in standard music encoding formats including MusicXML and MEI.

Contributions to Dezrann

Dezrann (Garczynski et al., 2022) is a score annotation browser application developed by the Algomus team and used in research and pedagogy for score visualization, annotation and analysis. In the frame of this research axis, Dezrann will be improved to provide a proper visualization and annotation of guitar tablatures. Although displaying tablatures in browser is already proposed by a number of commercial applications including SongSterr and Ultimate Guitar, Dezrann will to our knowledge be the first application to propose the adding of manual annotations, that can be visualized and exported for research and pedagogy purpose.

MuseScore plugins

The assisted-composition algorithms elaborated in the project will be implemented through user plug-ins which will enable the non-computer-scientist musician to use these methods through a comprehensive user interface. We plan to develop plugins implementing methods to assist the composition of rhythm guitar parts, bass tracks, to assist the adding of playing techniques and to assist the fingering choice to turn a generic score into a guitar tablature. Importantly, these tools will be used to evaluate the assisted-composition algorithms in the guitarist composer community.

The plug-ins will be integrated within the open-source music notation/composition software MuseScore. MuseScore's plugins are developed in the JavaScript based language Qt. Their creation is facilitated by a large and active developer community as well as a rich documentation[10]. The development of the plugin's interfaces will carefully follow findings from the composition practice survey (Section 3.4.2), especially ergonomic specifications indicating how the composers hope these tools to be controllable.

The software MuseScore is selected for this task for a number of reasons. It is free and open-source, and it is massively used by musicians across the world (7000 downloads per day in 2016[11]) with a foreground position in music pedagogy and musicology. The choice of MuseScore in this project is also strategic given the recent software's announcement regarding upcoming development efforts on guitar tablature processing[12] and more generally in composition features[13]. Crucial for this task, MuseScore puts substantial efforts in supporting a practical environment for the development of contributor's plugins enabling developers to implement specific functionalities into the software, which has already been used to experiment music algorithms (Hadjeres, Pachet, and Nielsen, 2017). Finally, as the tools targeted in this research axis are also intended to be used by non-guitarist musicians, the MuseScore community will hopefully be more impacted than user communities of guitar specific software such as Guitar Pro or Ultimate Guitar as this software is not restricted to the practice of any instrument in particular.

3.5 Impacts

Our research on tablature modeling has started in 2017 and our first articles on the topic were published in 2020. While it is too recent to enable us to estimate its impact to date on the scientific community, we hope that our work on tablature processing will soon become an essential part of the landscape of academic guitar research. We propose to detail in this section the expected impacts of the perspective research lines described in the last section.

3.5.1 Scientific impact

From the musicological point of view, we hope that the upcoming composition survey will improve our knowledge on modern popular music composition practices and encourage research initiatives in this understudied and protean area. More generally, the wish to address localized tasks within the composition process contributes to shift MIR research efforts in music generation towards human centered tools rather than systems emulating the composer's role (Esling and Devis, 2020). This point of view is in alignment with our recent work on musical co-creativity in the frame of the AI song contest (Micchi et al., 2021). In the MIR field, this research aims at promoting cutting-edge methods, notably involving machine learning, to address music analysis and composition problems that can be easily extended to

[10] https://musescore.org/fr/handbook/developers-handbook
[11] https://en.wikipedia.org/wiki/MuseScore#MuseScore_4
[12] https://musescore.org/en/node/326995
[13] https://musescore.org/fr/node/306609

other instruments than the guitar, in particular the piano which shares a number of properties. The concomitance of the PhD of L. Couturier on the classical piano repertoire (see Section 2.3.2) and the PhD of A. D'Hooge focusing on the modern popular repertoire are intended to benefit from each other and be pursued in close collaboration regarding the specific study of accompaniment textures, which presumably involves a number of common features between both repertoires. Finally, we hope that the open-source contributions (Music21, MuseScore and Dezrann) which are necessary for this project will enlarge perspectives of computational approaches to guitar music in the MIR community.

3.5.2 Economic impact

The musical marketplace includes numerous software dedicated to music composition and production, also known as Digital Audio Workstations (DAWs), such as Cubase, Live, or Logic Pro. The user can add specific functionalities and sound effects, also known as VST plug-ins, which has led companies to focus their activity on the creation of these platform-independent plugins to assist music composition and production. The implementation of algorithmic functionalities to assist music composition and production within DAW plugins has been experimented in late MIR research on the audio domain (Esling et al., 2019; Deruty et al., 2022) as well as the symbolic domain (Roberts et al., 2019) and prefigures a promising field of applications for companies developing DAWs and/or plugins as well as music notation software that aim at converging towards composition software[14]. The guitar is one of the rare instruments that has aroused the creation of specific notation/composition software, including Guitar Pro, Flat, Sibelius G7, probably because of the massive popularity of the instrument as well as its specific music notation system. Augmenting such software with composition functionalities similar to those proposed in this research axis opens a wide variety of commercial applications that should appear in the next years. Our discussions with Arobas Music (Guitar Pro) have indeed demonstrated a substantial interest of these companies in these innovative functionalities.

3.5.3 Social and cultural impact

The assisted composition tools of this research axis aim at facilitating the access to guitar music composition in the modern popular style to a wider circle of musicians. Music beginners, including non-guitarists, will get the opportunity to circumvent the lack of specific skills, that usually require years of practice, to compose realistic guitar tablatures. These tools therefore pursue at the level of the musical score a major technological shift that has contributed in the last decades to strongly facilitate the access to music production by the means of accessible and intuitive digital tools. We also estimate that these tools have the potential to encourage non-composer guitarists to experiment basic composition processes, and ultimately lead to unexpected composer vocations. On the other hand, the controllability of these tools will give the opportunity to expert musicians/guitarists to be exposed to unusual musical ideas, thus encouraging the diversification of their creative process. An important consequence of the adoption of these tools by the modern popular music composer

[14] MuseScore 4. Moving from notation software to composition software https://musescore.org/fr/node/306609

community would be to contribute to make evolve the relationship the guitarists have with the tablatures, not only as a memorization purpose but as a way to foster creativity while exploring new ideas.

Finally, a close collaboration and communication with the guitarist community will also be an opportunity to emphasize the potential of algorithm-based tools for the composers rather than for the audience, and to disambiguate possible ethical apprehensions regarding the principle of using algorithmic tools in composition (Ben-Tal, Harris, and Sturm, 2021).

Chapter 4

Questioning NLP approaches for score modeling

The research axis described in this chapter aims at studying the use of Natural Language Processing (NLP) methods to model musical scores. More specifically, we are interested in evaluating the adaptability, the performance and the limits of this increasing practice in the MIR community. This interdisciplinary research has been initiated in collaboration with M. Keller from the MAGNET team of the CRIStAL laboratory. It gave rise to the co-supervision of two Master internships (2021 and 2022) and one PhD (beginning in October 2022). This research axis is also the focus of a bilateral collaboration with D. Herremans from the AMAAI team at Singapore University of Technology and Design which is funded by a Campus France PHC project (2022-2023) that I am coordinating.

4.1 Computational processing of music *as* a language

4.1.1 Some modern NLP techniques

The field of Natural Language Processing (NLP) gathers a set of computational techniques intended to model natural language for a wide variety of applications including, among others, automatic text analysis, classification, translation, and generation.

Textual data are commonly structured as sequences of atomic elements such as sequences of characters or ideograms, and at some higher level sequences of words and sentences. Most NLP algorithms are then conceived to process sequences of elements, these elements being commonly referred to as *tokens*. Although tokens generally correspond to words or characters, a wide variety of *tokenization* strategies are discussed and compared in the NLP community (Mielke et al., 2021). We will see later in this chapter that tokenization seems to be a more critical step in the symbolic musical domain than in the text domain.

As an essential part of NLP, *language models* (sometime called *LMs*) are models that assign probabilities to sequences of tokens (Jurafsky and Martin, 2014). Forming an important part of the history of NLP research, *n-gram models* are one of the simplest type of language models. N-gram models are computed given a reference text corpus and allow to estimate the probability of a token given the previous ones and to assign probabilities to entire sequences. These probabilities are commonly

approximated using the Markov assumption which estimates that the probability of a token occurrence only depends on the n previous tokens.

NLP research has been considerably renewed in the last decade by deep neural network models, which have demonstrated unprecedented performances in a number of tasks. These models include recurrent networks such as Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), as well as attention based networks such as Transformers (Vaswani et al., 2017), which enable the measure of mutual relations between the words of a sentence. In contrast with recurrent neural networks, the transformer do not explicitly model information of token positions in its structure. Instead, it requires adding representations of absolute positions to its inputs, referred as *positional encodings*. Alternatively, *relative positional encodings* have been proposed by Shaw, Uszkoreit, and Vaswani (2018) to encode distances between tokens instead of their absolute sequence position.

The use of deep neural networks allow the design of particularly performant language models. Successive layers of deep neural networks allow the learning of concepts with increasing levels of abstraction, with the last layers ultimately aiming at modeling the meaning of entire sentences. When tokens correspond to words, the first layers in turn commonly aim at building expressive word representations commonly referred to as *word embeddings*. Embeddings represent words within sophisticated semantic spaces in which words with close meanings are represented by vectors with close values. Embedding methods are useful in wide variety of NLP tasks and have motivated the elaboration of dedicated algorithms such as Word2Vec (Mikolov et al., 2013). When applied to notably large and representative corpora, embeddings provide pre-computed word representations that can be re-used in concrete NLP tasks. This practice is a form of *transfer learning* (Radford et al., 2018), a practice that consists in pre-training a model on a general task with a large set of unlabeled data before fine-tuning it for the purpose of a specific downstream supervised task where a smaller set of data is available. Transfer learning methods also include the pre-training of deep neural networks, which have allowed major breakthroughs including GPT (Radford et al., 2018) and BERT (Devlin et al., 2019).

Although this variety of notions and techniques have originally been elaborated to address NLP tasks, their performance has encouraged their use in most research fields involving sequential data modeling, including in particular audio and music processing.

4.1.2 Using NLP techniques for symbolic music processing

Modeling musical scores with NLP techniques has a long history which seems to begin with the use of n-gram models (Brooks et al., 1957) dating back from the emergence of computer music. The elaboration of *viewpoint* representations (Conklin and Cleary, 1988) has thereafter enabled a wide diversification of musical n-gram models. Viewpoints provide sequential representations of a score excerpt with a focus on a combination of selected musical layers such as pitches, durations, intervals or melodic contours. They have been used for the tasks of music prediction (Conklin and Witten, 1995), pattern discovery (Conklin and Anagnostopoulou, 2001), genre classification (Conklin, 2013) and transformations (Bigo and Conklin, 2015). N-grams and viewpoints have also raised interest in the cognitive science field where they have been used to build music expectation models such as IDyOM (Pearce,

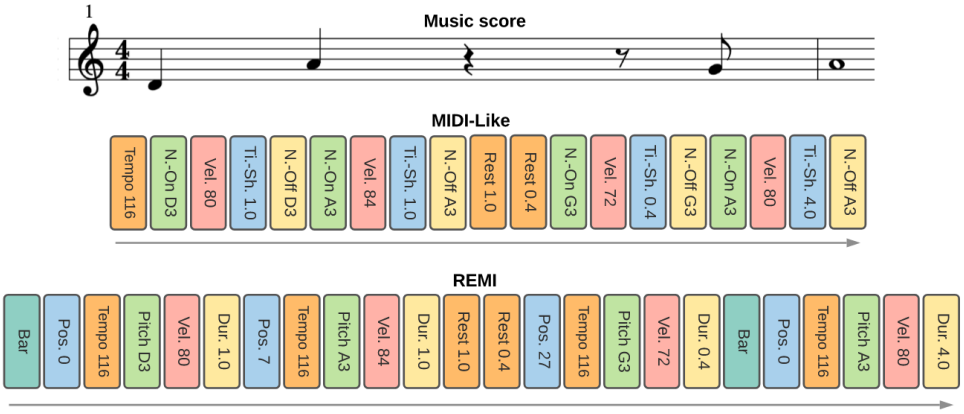


FIGURE 4.1: A score excerpt represented with two tokenization methods (Midi-Like and REMI). Figure extracted from (Fradet et al., 2021).

2005). At a lower representation level, string processing techniques such as factor oracles have in turn opened a wide research area in the field of machine improvisation (Assayag and Dubnov, 2004).

Over the past decade, the increasing use of deep neural networks such as transformers in music generation tasks has in turn promoted the elaboration of dedicated representation methods, based on sequences of musical tokens. The Midi-Like tokenization by Oore et al., 2020 reflects the syntax of MIDI messages (*e.g.*, `Note On` and `Note Off`) accompanied by specific tokens, `Time Shift`, which indicate the time interval between two successive MIDI events. In contrast, the REMI tokenization proposed by Huang and Yang, 2020, more closely translates the content of the musical score by introducing duration and position tokens exempting the use of `Time Shift` and `Note Off` tokens. Figure 4.1 illustrates these two tokenizations starting from a simple score excerpt. More recently, Hsiao et al., 2021 proposed a token vocabulary in which tokens describing a same musical event are grouped together as *compound words*. Compound words have a fixed length and their component tokens need to be processed by distinct heads of a transformer model. In order to facilitate the comparison of major tokenization methods in MIR research, the MidiTok python library developed by Fradet et al., 2021 allows the direct encoding of any MIDI content with most common tokenizations.

The influence of NLP in symbolic music representations has also led to research aiming at learning meaningful musical spaces analogous to word embeddings such as Word2Vec. These include Chord2vec (Madjiheurem, Qu, and Walder, 2016) and musical context vectors (Chuan, Agres, and Herremans, 2020).

Transferring NLP techniques to the musical domain has notably increased with the emergence of deep neural networks. Recurrent and transformer neural networks, which have originally been conceived for NLP, have become a standard tool for a number of MIR tasks, especially around music generation. The use of transformer neural networks for music generation has initially been introduced by Huang et al., 2019 with the *Music Transformer* model in which the transformer model, originally conceived for machine translation, is adapted to generate piano music. Importantly, this last study uses the principle of relative positional encoding which seems more adapted than the original positional encoding to model relative timing, which

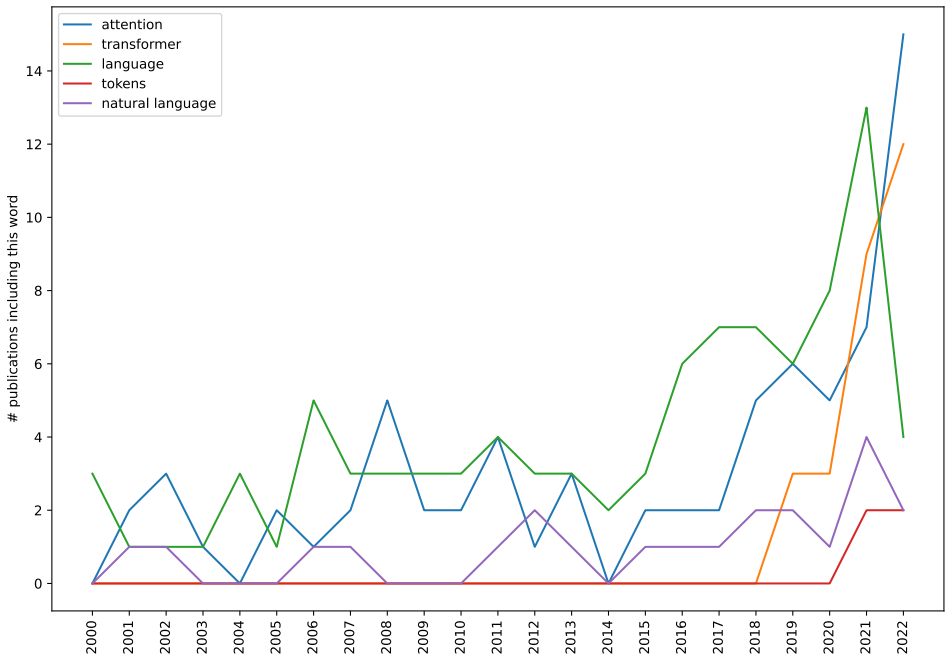


FIGURE 4.2: Number of occurrences of some terms related to NLP in ISMIR article abstracts, computed with the ISMIR explorer API (Low et al., 2019).

is an essential component in music. Aiming at facilitating the modelling of long sequences with recent linear-variants of the transformer, *Stochastic Positional Encoding* has recently been proposed as an alternative to relative positional encodings, showing promising effects on music generation (Liutkus et al., 2021).

Transformers have also been adapted in a variety of specific tasks including harmony analysis (Chen and Su, 2021), pop music generation (Huang and Yang, 2020), fingerstyle guitar tablature generation (Chen et al., 2020) or symphony music generation (Liu et al., 2022). Although transformer models are increasingly used in MIR, the transposition of the self-attention concept into the musical domain has rarely been studied to date. A system dedicated to music self-attention visualization has been proposed by the Google Magenta team (Huang et al., 2018) but without providing any tool allowing a systematic analysis of musical attention.

Figure 4.2 illustrates increasing occurrences of some words relating to the field within ISMIR article abstracts, including *transformer*, *attention* or *language*.

Inspired by its success in text with the BERT model, transfer learning has also been the object of a number of musical experiments lately. These include MuseBERT (Wang and Xia, 2021), MusicBERT (Zeng et al., 2021) and MidiBERT-Piano (Chou et al., 2021) which all aim at building unsupervised pre-trained models, which can be fine-tuned for musical supervised downstream tasks.

Interestingly, the possibility of using NLP tools in MIR also seems to have an influence on the formulation of MIR tasks. For instance, music generation driven by abstract feature scenarios, for instance emotion-based values, can be formulated as a Neural Machine Translation problem (NMT) where a model is trained to transform a sequence into another one (Makris, Agres, and Herremans, 2021).

Facing the increasing use of NLP techniques in the audio and musical domains, the workshop NLP4MuSA (Natural Language Processing for Music and Spoken Audio) has been created to gather research initiatives in this field. The NLP4MuSA workshop was organized as a satellite event of the ISMIR conference in 2020 and 2021[1].

4.1.3 Discussing the application of NLP methods to musical data

The application of natural languages techniques on musical data can be justified by a variety of arguments. We will mention three of them, accompanied by counter-arguments highlighting the need to perform this transfer of techniques with caution.

First, NLP and MIR share a number of *common tasks* including for instance content generation, style identification and style transfer. While these tasks have raised a number of works in both fields, their evaluation however presumably feature notable differences, essentially due to the subjectivity inherent to most musical tasks. While the evaluation of a number of NLP tasks also involves some subjectivity, the availability of a variety of benchmarks, which associate a dataset with a particular task, considerably facilitates the comparison of different representations and models. This is the case for example for the task of machine translation *e.g.*, French to English for which available benchmarks feature pairs of sentences in both language. Although more critical to evaluate, natural language generation can still be discussed on localized objective aspects such as grammatical correctness. However, evaluating music generation or style transfer arguably relies on even more subjective criteria including creativity and aesthetics, which complicates its automatization (Yang and Lerch, 2020).

Secondly, the temporal nature of music, similarly to speech, promotes its representation as *sequence of elements*, which happen to be the most commonly used data structure to represent text (*i.e.*, sequence of words, characters or ideograms). Melodies can for instance be abstracted by sequences of pitch values, and harmonic progressions as sequences of chord symbols. This similarity of data structures facilitates the application of NLP algorithms, and more generally of any sequence model, on simple musical data. Beyond these simple cases however, representing music as sequences of tokens appears to be much less trivial than text presumably because of the complex temporal organization of the elements forming a musical score. First, score elements in polyphonic music can overlap and occur simultaneously whereas textual elements, characters or words, are organized as pure sequences. As NLP models haven't been conceived to process any kind of simultaneity between events, a number of methods have been proposed to allow the representation of polyphonic music as purely sequential structures. Pitch slicing for example consists in segmenting the musical surface into temporal chunks labelled by the corresponding set of sounding pitches, which requires additional representations to distinguish notes that are attacked from notes that are held from the past, as in the vector representation discussed by Hadjeres, Pachet, and Nielsen, 2017. Representing polyphonic music by token sequences without any loss of information requires the use of time dedicated tokens, *e.g.*, time-shift tokens or relative score position tokens, which can not be structurally distinguished from pitch-based tokens by NLP models. Secondly,

[1] https://sites.google.com/view/nlp4musa-2021

score elements are associated with strict positions in time. While this notion is necessary to induce musical rhythm, it is absent in the text domain[2].

Finally, a frequent argument for the application of NLP methods to musical content is the *common association of music with a kind of language*. This assimilation is widely discussed in the musicology community (Cooke, 1990; Jackendoff, 2009). As pointed out by Jackendoff, a major difference is that natural language conveys propositional thoughts while music enhances affect, or emotion. It is worth highlighting that most modern NLP algorithms are based on deep learning architectures, whose conception were driven by the goal of extracting semantics from sentences, generally through a mechanism of progressive abstractions taking place in successive hidden layers of the network. Analogous abstraction mechanisms have shown to improve the modeling of music in a wide range of tasks (Briot, Hadjeres, and Pachet, 2019). It seems however that musical abstraction is unlikely to work the same way as semantic extraction for which such models have been originally designed, which questions in a general way the use of NLP tools to process musical data. This question can be illustrated with the application of self-attention in the musical domain. While the performance of self-attention in modeling natural language is often illustrated with its capacity to understand high-level grammatical concepts, the notion of grammar, as it is found in natural language, is unlikely to reflect the way music is structured. The nature of information that self-attention is able to model in music seems still unclear and relatively unexplored to date. While the next section details preliminary experiments on that topic, we hope to study this question further through the PhD of D.-V.-T. Le and our starting collaboration with D. Herremans at AMAAI (Singapur) who recently had promising results on experiments aiming at transposing the self-attention mechanism in the pitch domain (Guo, Kang, and Herremans, 2023).

4.2 Contributions

The research project described in this chapter is more recent than those described in Chapter 2 and 3. It started in 2019 and has led to three publications, all resulting from student works that I have been co-supervising with M. Keller. These publications respectively focus on musical context vectors (Keller et al., 2021), musical self-attention interpretation (Keller, Loiseau, and Bigo, 2021, Section 4.2.1) and musical tokenization (Kermarec, Bigo, and Keller, 2022, Section 4.2.2).

4.2.1 Interpreting self-attention in musical scores

Following research lines focusing on the effect of the self-attention mechanism on musical data mentioned in Section 4.1.2, the present work aims at opening the Music Transformer *black box* and evaluate the ability of the self-attention mechanism to convey high-level musical information.

Following NLP initiatives aiming at studying self-attention (Reif et al., 2019) we isolated self-attention information and submitted it to two selected MIR *probing tasks* : composer classification and cadence detection. Figure 4.3 illustrates this process.

[2]Interestingly, the notion of rhythm commonly appears in poetry, which can in many senses be considered somewhere in between music and natural language as highlighted by Jackendoff, 2009

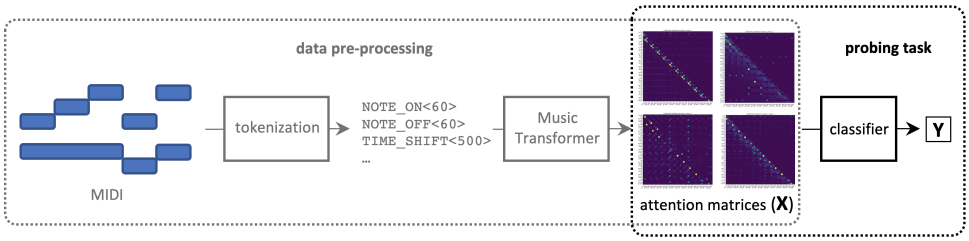


FIGURE 4.3: Pipeline used for the two probing tasks. The left part illustrates the systematic representation of a MIDI sequence into a set of self-attention values computed by a music transformer trained on the MAESTRO dataset. The right part illustrates how a probing task is formulated as a classification problem from attention values (Keller, Loiseau, and Bigo, 2021).

First, a music transformer model (Huang et al., 2019) is trained on a dataset of musical works in MIDI format, the MAESTRO dataset (Hawthorne et al., 2019), which have been represented as sequences of tokens. Once trained, any musical sequence given as input to the model will give rise to the computation of a set of pairwise self-attention values which are stored within self-attention matrices at each layer of the network. This process can then be used to systematically represent a whole musical dataset by abstract self-attention matrices. A logistic regression classifier was then trained to estimate the composer of a set of musical sequences represented by their self-attention matrices. We naturally do not aim at being competitive with state-of-the-art composer classification methods, but rather measure the ability of self-attention values to carry style information.

Figure 4.4 displays the accuracy of several binary composer classifiers for different size of sequences. The pairs of composers have been deliberately selected to illustrate cases of variable difficulty. For instance, J. Haydn and W.A. Mozart are known to be close in style, which make their distinction complex. In contrast, couples such as J.-S. Bach and F. Chopin or W.A. Mozart and C. Debussy are much easier to separate. The results show that self-attention seem indeed to carry stylistic information as shown by the easy couples having an accuracy above 0.5. Self-attention values do not seem however to be able to distinguish difficult cases such as J. Haydn and W.A. Mozart, independently of the size of the sequence.

As another illustration, Figure 4.5 displays the cumulated self-attention computed at each sixteenth note of a score excerpt featuring a perfect authentic cadence. The distribution exhibits attention peaks at strong preparation points of the cadence, which seems to indicate the ability of self-attention to model some high-level structural information in musical scores. This is confirmed by a cadence detection experiment on an annotated corpus of fugues from J.-S. Bach (Giraud et al., 2015). A logistic regression classifier is trained on self-attention matrix representations, similarly to the experiment on composer classification, and obtains an accuracy 15% above the random classifier.

4.2.2 Investigating the expressiveness of tokenizations

Most strategies representing score content as sequences of tokens commonly encode pitch information explicitly, for example with tokens such as `pitch:C3`, which

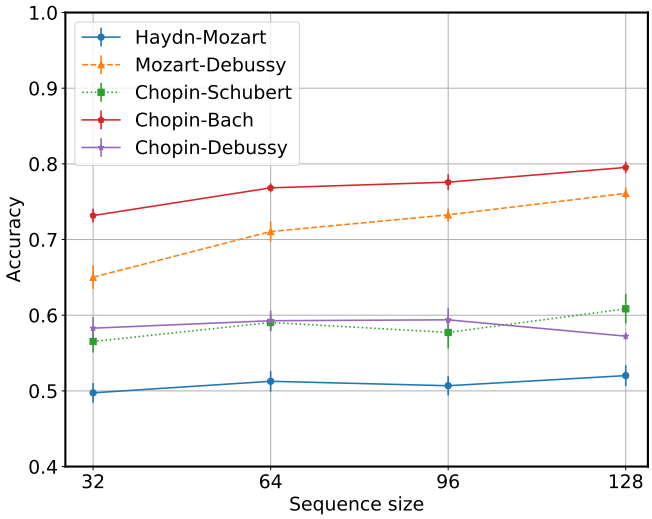


FIGURE 4.4: Binary composer classification performed on self-attention matrices computed with the music transformer trained on the MAESTRO dataset (Keller, Loiseau, and Bigo, 2021).

are afterward fed to a sequence model. This contrasts with a natural tendency of human listeners to perceive and memorize musical sequences in terms of relative pitches. This difference tends to limit the ability of machine learning models to exploit learned musical knowledge across different keys. This problem is generally tackled by a data-augmentation procedure that consists in applying various transpositions to the training data, which consequently increases the resources required to train the model. As an alternative, this work experiments a transposition invariable token representation which encodes pitch intervals and facilitates the uniform transposition of musical knowledge learned by sequence models at any key without any resort to data augmentation.

Figure 4.6 illustrates three tokenizations of a short musical excerpt. The top line corresponds to the REMI tokenization (Huang and Yang, 2020). In the second one, pitch tokens are replaced by pitch interval tokens, making this token sequence transposition invariant. The tokenization of the bottom line distinguishes pitch-interval tokens between notes with consecutive (*Horizontal Pitch Interval, HPI*) and simultaneous onsets (*Vertical Pitch Interval, VPI*).

The expressiveness of these transposition invariant tokenizations have been studied in the frame of a binary composer classification task and an end-of-phrase detection task. The GiantMIDI-Piano dataset (Kong et al., 2022) was used to train and evaluate the composer classification model and the TAVERN dataset (Devaney et al., 2015) was used for phrase end detection. We used logistic regression classifiers on musical sequences represented as *bag-of-tokens* with TF-IDF weights[3]. Figure 4.7 compares the accuracy obtained by a set of logistic regression classifiers trained to these tasks with various tokenizations.

[3]term frequency-inverse document frequency : counting the number of occurrences of each token in the sequence and scaling the count by the frequency of the token in the corpus.

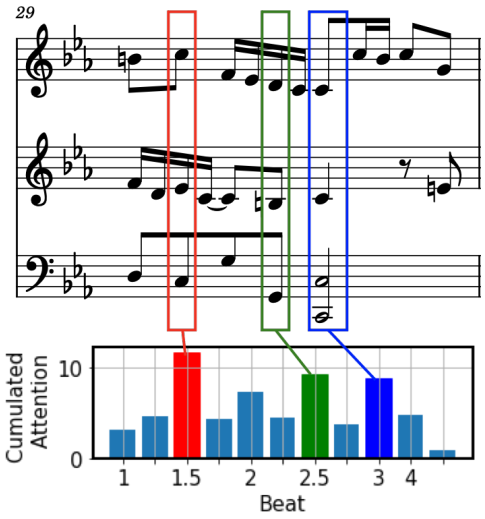


FIGURE 4.5: Cumulated attention on successive offsets of bar 29 of Fugue 2 of the *Well-Tempered Clavier* from Bach. A perfect authentic cadence is annotated on beat 3 (last frame, blue). Other points of prominent attention (first two frames, red and green) correspond to important *preparation* points of the cadence (Keller, Loiseau, and Bigo, 2021).

The differences of accuracies show that the tasks vary in difficulty and that the choice of tokenization can have dramatic impacts on the performance of the classifiers. REMI and Pitch Interval tokenizations have comparable performances for composer classification, except for distinguishing F. Schubert and F. Chopin where Pitch Interval tokenizations perform better. We hypothesize that the performance of REMI for the two other classifications is partly due to the pitch range difference between the repertoires of the composers, F. Liszt and L.v. Beethoven arguably employing larger pitch ranges than Bach and Mozart, which is by nature better encoded by absolute pitch tokens. Finally, we see a significant out-performance of the spatial pitch interval tokenization (that distinguishes horizontal and vertical pitch interval tokens) for the end of phrase detection, presuming a promising ability of this representation to model abstract musical knowledge.

4.3 Impacts

The NLP field is likely to show major progress in the near future, and to provide innovating algorithms that will in turn be experimented in the musical domain, pursuing the tendency observed in the MIR community in the last decade. This research axis is based on the hypothesis that the popularity of NLP models in the MIR field is more due to their impressive performance on natural language than by an obvious parallel between music and natural language[4]. One ambition of this project is therefore to encourage a reasoned use of NLP methods within our musical community. Given the complexity of NLP-dedicated neural networks, we believe that a better

[4] A similar hypothesis could probably be formulated to comment the popularity of techniques coming from image processing, such as convolutional neural networks, in the MIR community although music and images strongly differ in nature.

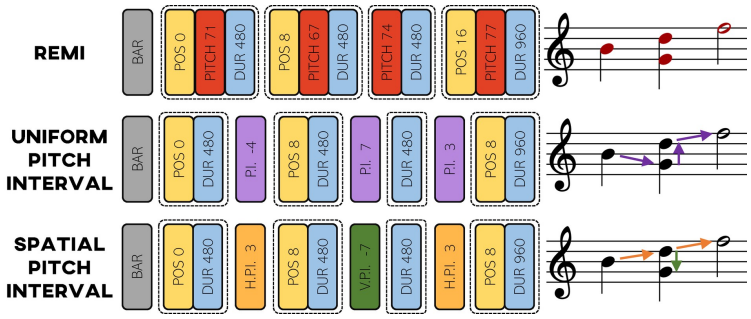


FIGURE 4.6: Three tokenizations of a musical sequence. The dotted frames group tokens describing a same note (Kermarec, Bigo, and Keller, 2022).

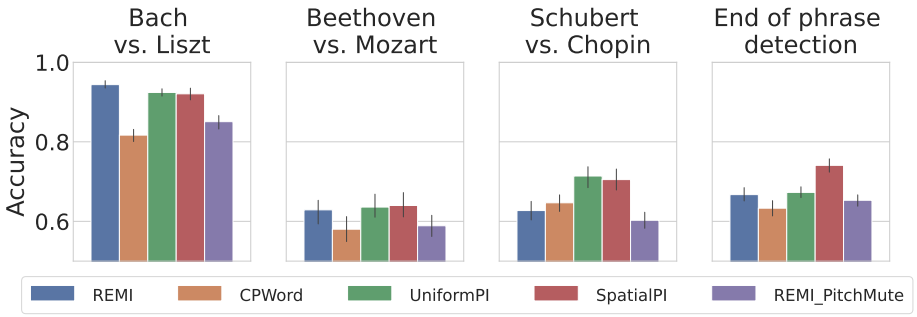


FIGURE 4.7: Composer classification and end of phrase detection performed by a logistic regression on TF-IDF token representations, with 5 different tokenizations (Kermarec, Bigo, and Keller, 2022).

understanding and control of these models can contribute to limit unnecessary expensive training procedures and limit energy consumption that is now recognized as a major issue in machine learning. This goal meets a global initiative aiming at privileging light audio and music processing models (Douwes, Esling, and Briot, 2021). Using musically expressive representations of data as proposed in Section 4.2.2 aims at limiting the volume of training data necessary to set up MIR models, therefore reducing time and energy consumption. Improving our understanding on how self-attention behaves within musical data, as proposed in Section 4.2.1, promotes a use and setting of transformer models with better intuitions regarding their capacities, and therefore make research experiments converge towards accurate results with limited experimental trainings.

Deep neural networks dedicated to NLP generally aim at modeling semantic concepts through progressive layer abstractions. Using such network architectures to model high-level musical concepts is therefore based on the questionable assumption of analogous abstraction mechanisms between text and music. By trying to identify limits of transferring techniques from NLP to MIR, we hope to propagate the challenging idea that the design of music modeling neural architectures could be driven by the very nature of music instead of being adapted from architectures that have shown success in widely different domains, such as natural language or image. Music models could be inspired by NLP models without necessary imitating them.

As mentioned in the motivation of this research axis, we hope in the long-term that this work will contribute to a general reflection around parallels between music and language, that already has a long history in humanity fields relating to natural language and musicology (Cooke, 1990; Jackendoff, 2009), and gain a clearer idea of what the term "musical language" could mean.

4.4 Perspectives

The future directions of this research axis will be first addressed in the PhD of D.-V.-T. Le that I am co-supervising with M. Keller. The interdisciplinary of this topic encourages us in a first place to establish a crossed state-of-the-art gathering NLP techniques, MIR tasks and symbolic music representations, as well as an overview of how NLP techniques have been adapted in the musical domain to date. We hope this overview can reach to a general survey paper on the uses of NLP techniques to model music. We will then focus on the following specific tasks.

4.4.1 Tokenization strategies

Our research on tokenization has begun with the internship of M. Kermarec and is continued by the PhD of D.-V.-T. Le. This topic is also central in the PhD of N. Fradet (LIP6, Sorbonne Université) with whom a promising connection is currently being established. This collaboration is currently facilitating the use and contribution to the MidiTok python library, dedicated to music tokenization (Fradet et al., 2021).

Improving token type selection

The work done by M. Kermarec described in section 4.2.2 contributes to a wide opening research area on the problem of musically expressive tokenization. A variety of additional strategies are planned to be experimented to address this task.

The experiments by M. Kermarec showed that transposition invariant tokenizations, although promising for the generalization of musical knowledge regardless of the key, brings limitations in style modeling due to the complete loss of register information. To circumvent this limitation, a first perspective consists in adding periodic octave information tokens, for instance at every bar, thus providing an insight to the model about the register of the surrounding notes.

Another alternative would consist in using a hybrid tokenization where a selected set of notes would be encoded with their explicit pitch values while some others would be encoded using interval tokens relatively to the first ones. The spatial pitch interval tokenization presented in Section 4.2.2 provides a natural way to perform such a distinction. Using only vertical pitch intervals to encode harmony notes and preserving explicit pitch values for top notes would inform the model about pitch register while allowing the learning of harmonic patterns up to transposition. Although this approach would limit the modeling of melodic features up to transposition, it could presumably bring perspectives for harmonization tasks. The choice of such a compromise remembers us the necessity to keep in mind targeted applications when designing a model.

While most of our tokenization perspectives so far have been thought to improve pitch representation, music tokenization also faces challenging limitations regarding the encoding into simple raw sequences of complex temporal information inherent to music, such as precise timings and simultaneity. While not having yet precise intuitions on how to improve these aspects, we hope this project will contribute to address them.

Statistical *super-tokens*

Beyond the selection of a set of expressive tokens to form a vocabulary, the NLP field has elaborated techniques consisting in enlarging vocabularies with *super-tokens* that correspond to groups of consecutive tokens that frequently occur in a reference corpus.

Given a reference corpus tokenized with one of the above methods, the byte-pair encoding (BPE) algorithm by Sennrich, Haddow, and Birch, 2015 enumerates the k most frequent token n-grams in this corpus, add them as super-tokens into the basis vocabulary, then replace the corresponding n-grams by these super-tokens in the corpus. As k grows, the BPE algorithm is likely to produces longer super-tokens, which will make the vocabulary becoming more and more expressive, but also more and more dependent on the reference corpus. At the training stage, this dependency is likely to limit generalization capacities as well as creative outputs in the case of a generative model. The choice of k will naturally depend on the model application which is foreseen, but also on the size and variety of the reference corpus. Although BPE seems to start drawing attention in MIR community as shown by the work of Liu et al., 2022 as well as announced in upcoming improvements of the MidiTok library[5], its effects specifically on musical data has still not been the subject of any study to our knowledge.

As an alternative to the BPE algorithm, the WordPieceModel algorithm by Schuster and Nakajima, 2012, selects super-tokens depending on their abilities to optimally improve the likelihood computed on the reference corpus encoded with these super-tokens. While this method has been successfully applied in the field of speech recognition, it hasn't been experimented yet in the music domain. This part of the PhD of D.-V.-T. Le will be facilitated by collaborations in the Magnet team, which maintains the NLP `mangoes` library[6] dedicated to embeddings building.

Words and music

Most NLP algorithms are designed to process sequences of tokens that in most time happen to correspond to sequence of words. Segmenting English or French texts into sequences is relatively convenient given the natural spacing between words in text[7].

On the musical domain, most uses of transformer models are using tokenizations, such as Midi-Like or REMI, in which tokens represent information at a level comparable to that of MIDI events. Comparing tokenized text and tokenized music

[5] https://github.com/Natooz/MidiTok

[6] https://gitlab.inria.fr/magnet/mangoes

[7] Although specific rules are required to process special characters such as hyphens and apostrophes

highlights a gap of expressiveness between text tokens and music tokens as an isolated word in text has probably good chances to convey more information than an isolated MIDI event in a musical sequence. This contrast is surprisingly rarely mentioned in MIR despite an increasing use of transformer models processing Midi-like tokens.

In this sense, we believe that the parallel between music and text might benefit from comparing MIDI events to characters rather than words. Interestingly, NLP includes some research initiatives aiming at tokenizing text at the character level (Mielke et al., 2021) giving the potential to a model to generate any kind of words, including words absent from the training corpus. Such generalization ability might for instance let the model generate the unseen word `faster` which will be deduced from occurrences of `fast`, `strong` and `stronger`[8]. The natural counterpart is that character level models are likely to produce nonexistent words. Modeling text at the character level is particularly relevant for some Asian languages that have no or few spaces between words. The elaboration of the WordPieceModel algorithm mentioned above has actually been motivated by the need to segment Japanese and Korean texts into sequences of tokens. We hope this project will contribute to encourage careful applications of "word models" on data which could hardly be compared to words, and therefore promote the adaptation of character-level approaches that seem more convenient to process musical information, at least when it is represented by tokens at the level of MIDI events.

4.4.2 Transfer learning

Transfer learning consists in pre-training a model on a general task with a large set of unlabeled data before fine-tuning it for the purpose of a specific downstream task where a smaller set of data is available. This technique has enabled major breakthroughs in NLP with the pre-training of models such as BERT (Devlin et al., 2019) or GPT (Radford et al., 2018). Transfer learning has been recently investigated in symbolic music data processing (Wang and Xia, 2021). Starting from this recent research, we hope to measure the benefit of transfer learning in different tasks in symbolic music analysis and generation.

Transfer learning is based on the hypothesis of common concepts shared by data of a domain. It can for instance benefit to a set of texts that are written in the same natural language and in which concepts such as grammatical relations will be appliable for any data. It can also benefit to a set of images that tend to describe the same world, composed by objects that will ultimately be grouped under abstract classes, such as animals or vehicles. Applying transfer learning to music questions us on what is shared by musical works. Distant musical styles, say for instance twelve-tone and classical music, might probably share too little in common to jointly facilitate the learning of a universal sense of music, comparable to what is performed in image processing for instance. Musical transfer learning therefore contributes to question us on the existence of a *musical language*.

[8] This property can also be achieved with word level tokenizations using token segmenters such as Byte-Pair Encoding.

Chapter 5

Conclusions

Music begins where the possibilities of language end.
— Jean Sibelius

The perspective of my different research axes have been detailed in the closing sections of Chapters 2, 3 and 4. This conclusion chapter first summarizes the research described in the manuscript. We will then discuss the analogy between music and language which was used as common thread in the manuscript, and put this analogy in parallel with the use of symbolic spaces in music theory. The last section discusses possible future evolutions of the role of AI in music composition.

5.1 Summary

This section has been written with the help of ChatGPT 3[1]*, which was asked to translate and summarize the French content written on page* vi *of this manuscript. This experiment illustrates the "punctual" use of an AI tool in the writing of a document, in analogy with the composition of a musical piece, as discussed further in Section 5.3.*

This manuscript described my main research axes around the general theme of modeling the language of musical scores, in the field of Musical Information Retrieval (MIR). The manuscript was organized around three chapters, respectively dealing with classical repertoire, guitar tablatures in modern popular repertoire, and the application of Natural Language Processing techniques in the musical field.

Chapter 2 focused on algorithms to analyze classical music. It covered PhD works of L. Feisthauer and L. Couturier on the modelling of music structure and texture that I have co-supervised. I presented a method for identifying cadences in sheet music using a set of descriptors. The second research axis focus on the modeling of symbolic texture in sheet music for piano. Finally, I presented methods for the structural segmentation of sheet music according to the general scheme of Sonata Form. The perspectives of this axis mainly focus on texture modelling in the frame of the PhD of L. Couturier, in particular formalizing distances between textures and using these models for computer-assisted composition and style transfer.

Chapter 3 focused on the modelling of guitar tablatures in computer-assisted music analysis and composition in the popular modern repertoire. The chapter highlighted the internship work of J. Cournut and D. Régnier, as well as the present PhD work of A. D'Hooge that I have been, and that I am co-supervising. This research

[1] https://openai.com/blog/chatgpt/

is part of the ANR project TABASCO that I am coordinating. The chapter detailed the development of tools to aid in the use of MySongBook data for machine learning tasks, as well as methods for automatic identification of the musical function of guitar tabs, and a method for continuing rhythmic guitar textures through texture imitation. The research perspectives in this axis pertain to the ANR TABASCO project, which aims to develop AI-based tools to assist in composing music for guitar. It will include a study of guitar composition practices in modern popular music, conducted in collaboration with musicologist B. Navarret. The goal is to improve our understanding of the role of notation software in the composition process in this repertoire and identify software features that could lead to innovations in composition practices. The project will then focus on creating and evaluating these features, with a particular emphasis on composing rhythmic guitar parts and modeling guitar techniques to provide fine control over the expressiveness of the music. The project will also include open-source contributions to make it easier for the MIR community to work with guitar tablatures.

Chapter 4 discusses the use of Natural Language Processing techniques in modeling musical scores. The study is motivated by the increasing use of algorithms designed for text in music research and aims to evaluate the relevance, potential, and limitations of bringing these two fields together. The first contribution of the chapter is the study of how the principle of mutual attention applies to musical scores, and the second contribution is about tokenization which is a method of representing musical scores in the form of a sequence of atomic elements to allow the direct application of NLP algorithms such as the transformer. The study also includes an evaluation of token expressiveness through two musical tasks: composer classification and cadence detection. Perspectives of this research axis include the identification of a map of uses in this field and to focus on the sequential representation of musical content, tokenization and the use of recent algorithms in music, such as WordPiece and Byte Pair Encoding. The project also aims to experiment with the limits of transfer learning in music and contribute to a general epistemological reflection on the similarities and differences between music and natural language. The research perspectives of this project mainly correspond to the axes anticipated for D.-V.-T. Le's thesis.

5.2 Music, language, space, computer science

The inherent complexity of music seems to encourage theorists to refer to intuitive concepts which are not necessarily relating to music in the first place. The language analogy was used as a common thread to present various research in this manuscript. Interestingly, natural language might not be the only abstract notion used by music theorists for this purpose. This section puts in parallel the notions of *language* and *space*, which are metaphorically, although differently, employed by theorists to bring some intuitions in the description of some complex components of music such as expressiveness and harmony. My intuition is that the use of such analogies reveals our lack of terms to describe the complexity of music, of its structure and of its effect on the listener.

5.2.1 Music and language

While the language analogy is an essential component of our reflections on the use of NLP techniques for music (Chapter 4), the very notion of language is presumably playing a much less central role for the research focusing on the classical repertoire (Chapter 2) and on guitar tablatures (Chapter 3). Pursuing this analogy was however possible due to the various and endless parallels that can be drawn between the two domains.

Music indeed seems to be associated with the notion of language for a variety of reasons. In the common sphere, this association for instance refers to a transmission act from the composer to the listener, or simply the fact that a music can potentially be appreciated in the same way by people from different cultures or languages. Below a general view of music as a language, musical data, whether from scores or from performances, include a variety of concrete features that can easily be put in parallel with linguistic features. Musical *cadences* for instance are often described as formula indicating *end of phrases* in musical scores. The notion of musical language also questions cross concepts such as musical syntax or musical grammars, that have led to reference theories such as the *Generative Theory of Tonal Music* by Lerdahl and Jackendoff (1983).

But as mentioned in Section 4.1.3, natural language and music also have fundamental differences, one of which being that the former conveys "propositional" or "conceptual" thoughts (Jackendoff, 2009), while the latter is commonly described as expressing emotions (Cooke, 1990), or more generally affects (Jackendoff and Lerdahl, 2006). In his book *The Language of Music*, Cooke (1990) suggests that associating music with a language helps us to share our understanding of the artist's intention[2].

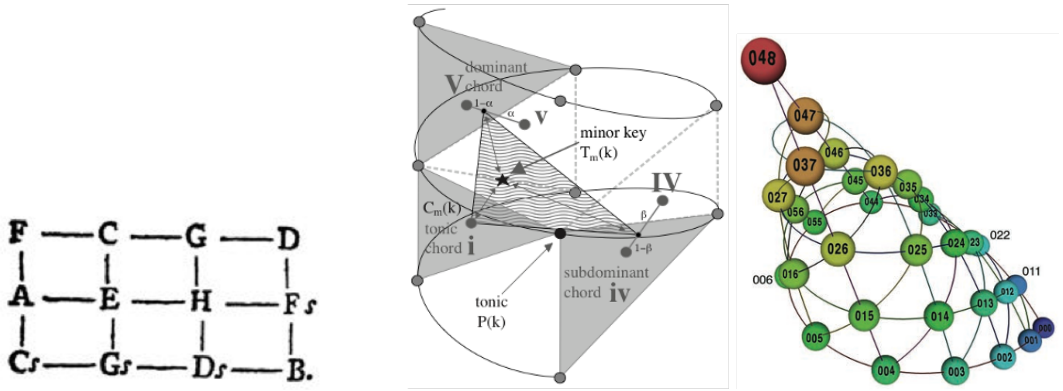
In the next section, I'll mention the use of another intuitive concept in music representations, the notion of *space*, which was the focus of my PhD thesis (Bigo, 2013).

5.2.2 Music and space

A wide variety of spatial representations have been elaborated by music theorists to facilitate the description of complex aspects of music such as tonality and chord relationships (Mazzola, 2012; Tymoczko, 2010). Figure 5.1 illustrates three such representations. The Tonnetz, first described by the mathematician Euler (1739), represents acoustic pitch consonance by graph neighborhood. The spiral array model of Chew (2000), includes concentric helices highlighting in a same space relations shared by pitch-classes, triads and keys. The orbifold $\mathbb{T}^2/S_3$ by Callender, Quinn, and Tymoczko (2008) represents voice-leading relations between three-note chord types in a cone.

Such spatial representations have provided tangible intuitions on a variety of tasks in music theory, analysis and composition. A movement in a space, which is a concept we are confronted with every day, is indeed easier to grasp than the physics and combinatorics that characterize pitches and chords. Intuitive representations

[2]Cooke actually generalizes this hypothesis to analogies between arts in general.



(A) (B) (C)

FIGURE 5.1: Three uses of space to represent a complex aspect of music. (A) The *Speculum Musicum* representing physical consonance. (B) The *Spiral Array* representing relations between pitch-classes, triads and keys. (C) The orbifold $\mathbb{T}^2/\mathcal{S}_3$ representing voice-leading relations between three-note chord types. Figures from Euler (1739), Chew (2000) and Callender, Quinn, and Tymoczko (2008).

also promote original experiments leading to unexpected results. For instance, associating a musical sequence with a spatial trajectory within expressive music representation spaces suggests some natural transformations such as rotations and translations, that provide fruitful and original approaches to composition (Bigo et al., 2015). As a second example, representing pitch-class set collections as geometric simplicial complexes has incited us to apply the principle of filtration, inspired from the field of persistent homology in mathematics, to bring some original approaches to music similarity (Bigo and Andreatta, 2019). Figure 5.2 illustrates the first bars of the four-voice chorale BWV 254 from J.-S. Bach. The left most box indicates that the pitch-class F is the only one to sound more than 60% of the time in this extract. Similarly, the second box indicates that {F,A} is the only pair of pitch-classes that sound simultaneously more than 45% of the time. Preliminary experiments seem to show that filtration levels and their associated pith-class content, considered up to transposition, highlight some style related information (Bigo and Andreatta, 2019). I hope this approach could bring a promising point of view for the processing and analysis of similarity at different scales of representations, which is the central question of the MUSISCALE action of the MADICS GDR in which I am involved.

I will conclude this parenthesis on spatial representations of music by highlighting that although music relates to the concepts of space and language in rather different ways, some musical features that are commonly used to justify the linguistic aspect of music, for example keys and chord progressions (*tonal language* and *harmonic language*) are interestingly also described through well-defined geometrical spaces (*key spaces* and *chord spaces*) due to their combinatorial properties.

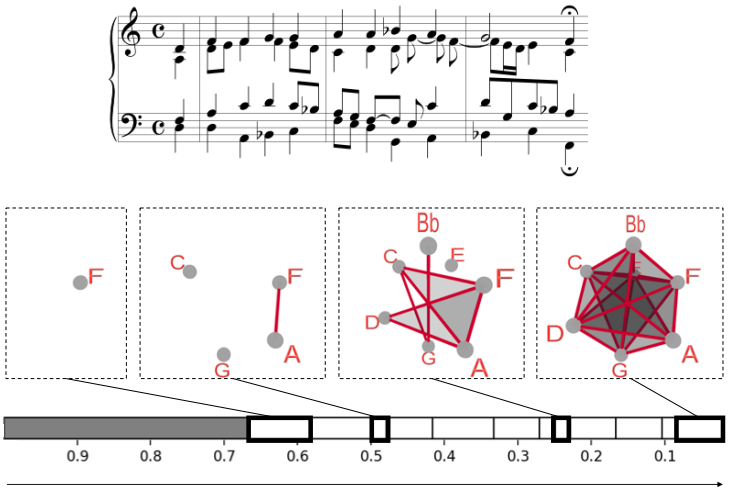


FIGURE 5.2: Representation of four levels of pitch-class set filtration in an extract of a four-voices chorale by J.-S. Bach. On the left side, only the most prevalent pitch-class sets are represented. The most right level includes the whole set of pitch-class sets included in the extract.
(Bigo and Andreatta, 2019)

5.2.3 Abstract analogies and dedicated computer science

While used in different contexts, both previous analogies have in common to open deep perspectives in music modeling when approached with computational methods, thanks to the availability of powerful computational tools dedicated to both concepts. On the one hand, the field of *spatial computing* (Giavitto et al., 2004) aims at taking advantage of our intuitive relation to space, to provide a framework to formulate complex data structures and algorithms in spatial terms with a number of applications to process biological (Giavitto and Michel, 2003) and musical (Bigo, 2013) data. On the other hand, the field of Natural Language Processing provides tools to process and analyze large amount of natural language data, with a variety of applications ranging from automatic translation to text generation (Jurafsky and Martin, 2014). Spatial computing and Natural Language Processing allow the application to musical data of powerful algorithms, although initially thought for different kind of data and therefore provide original approaches to music analysis and composition.

The availability of powerful computational tools can however make it tempting to simplify the complexity of musical information to facilitate their application. Limiting these simplifications and keeping in mind the singular nature of music when using these tools brings promising challenges in our research area. Pursuing the reflection on the use of powerful computational tools in the musical domain, the next section discusses the use of artificial intelligence tools in music composition.

5.3 Human-centered computer music

Artificial intelligence has a growing impact on multiple aspects of our daily lives and already succeeds in replacing the human in some tasks such as driving a car or

identifying faces on a picture. Such breakthroughs legitimately lead us to question the potential of this technical shift to converge towards the replacement of human in other tasks, including in the musical domain. In the context of music composition, trying to presage future impacts of AI seems however complicated due to the variety of contexts in which music is involved nowadays. In this closing section, we discuss AI music composition by distinguishing *functional* and *unfunctional* music.

5.3.1 *Functional* music

AI techniques have already succeeded in replacing the human composer for a particular category of music, which is intended to be played in the *background* of some stream videos[3] (documentary, tutorials, humorous, advertising), in which music with limited creativity sometimes seems acceptable due to its background role. This of course only concerns a part of this type of videos, as documentary and advertising music can also arouse the composition of amazing pieces of arts. For instance, the Jukedeck[4] company used to provide copyright-free songs generated at the click of a button, only requiring from the video maker to select a musical style and a duration. Jukedeck generated pieces have successfully been adopted as backtrack music of YouTube videos by a large community of video makers.

An important aspect of music in this category is that it seems expected to fulfil a precise *function*[5]. I believe that this notion of function fulfillment plays an essential role in the success of AI to autonomously generate such music. First, the goals of functional music are not necessarily related to the music itself, which presumably makes it less demanding in terms of creativity. This probably also contributes to make the absence of human in its composition more acceptable by the audience, which in most cases will not even be aware of this fact. Finally, the generation of functional music seems more compatible with well-defined problem-solving tasks for which machine learning algorithms have originally been conceived for. This generally questions the ability of AI to be *creative* rather than *performant* and which is widely discussed in the community (Esling and Devis, 2020; Gioti, 2021).

Pursuing this intuition, I feel that the audience is likely to welcome in the next years generated music addressing other types of expectations associated with specific listening contexts such as sport performance, intellectual concentration, relaxation, health therapy, and to some extent dancing although this last objective seems more challenging[6].

5.3.2 *Unfunctional* music

On the other hand, the idea of going to a rock/classical concert in which the performed songs/pieces would have been composed without any human intervention seems much more unlikely to me, primarily because this might not be of interest of the (human) audience, however credible the generation might be[7]. My feeling is that

[3] For example: https://www.youtube.com/watch?v=T7RLgpwwRsg.

[4] Jukedeck has been bought out by ByteDance (TokTok) in 2019.

[5] The term *functional music* was defined by Gaston (1958).

[6] It might be more likely to presage that generated dance music will arouse new dancing styles rather than being used in existing ones.

[7] This might however be different is the audience is not *aware* that the composition did not involve any human intervention.

this category of music, which is composed for entertainment and pleasure and that we could qualify as *unfunctional*, will continue to require a human "in the command" of its creation process to reach a substantial audience. This also naturally applies to a part of functional music discussed in the previous paragraph.

Despite this presumed necessity of a human control at the top of the process, I nevertheless believe that AI is going to play increasing, various, and probably unexpected, roles within the act of composing *unfunctional* music. While composition processes arguably vary across composers and musical styles, it seems reasonable to consider that they often involve a set of more or less ordered and interacting subtasks such as for instance melody finding, harmonization, orchestration, notation or mixing. My intuition is that AI approaches are particularly promising when limited to one of these tasks. First, in contrast with *full-stack* generation, composition subtasks seem more easy to formalize as computational problems, and also to evaluate, as they are generally more precisely defined. The rhythm-guitar tablature continuation by texture imitation described in Section 3.3.5 is an example of well-defined problem whose formalization and evaluation appear much better determined than the open-ended creation of a whole musical piece. Secondly, addressing a composition subtask might presumably benefit from results in computational music analysis, which are generally limited to specific musical aspects such as musical texture as described in Section 2.3.2. Finally, punctual AI interventions appear promising in situations where the composer does not feel equally skilled, inspired, or simply interested, in each of these subtasks.

Our participation to the AI Song Contest in 2020 (Huang et al., 2020) follows this line by addressing the composition of a whole musical piece as a combination of subtasks alternatively involving the human and the computer (Micchi et al., 2021). This experiment however showed us that addressing punctual composition subtasks brings the challenge of integrating these tools within the whole composition process (Deruty et al., 2022). Increasingly sophisticated Digital Audio Workstations (DAWs) and music notation software, which have become central tools in many composition acts, seem to have the potential, in terms of techniques and interfaces, to address this challenge. This will however require substantial composition practice studies aiming at identifying the precise form under which such tools should be made available to the composer during its composition act.

To conclude on punctual AI interventions, I will somehow rudely attempt a parallel between the composition of a musical piece and the writing of a technical document such as the present one, in which the different sections naturally vary in terms of creativity. Section 5.1 is probably one of the less creative sections as it aims at summarizing the information that has been previously detailed in the manuscript. The AI tool ChatGPT was therefore used for translation and summarization of the French summary of page vi, after which some human post-processing was done to correct possible mistakes and make the summary fit better into a conclusion chapter. This post-processing step, which consists in correcting and integrating the generated content in a wider context, could easily be compared to what a composer could do with a generated melody as described for instance by (Ben-Tal, Harris, and Sturm, 2021). Using ChatGPT to assist the writing of a committed discussion section, as the present one, seems however more critical unless the proposed ideas have been mentioned in the past.

List of publications

Agres, Kat, Louis Bigo, and Dorien Herremans (2019). "The impact of musical structure on enjoyment and absorptive listening states in trance music". In: *Music and Consciousness 2 : Worlds, Practices, Modalities*. Oxford University Press. URL: https://hal.archives-ouvertes.fr/hal-02024603.

Agres, Kat, Louis Bigo, Dorien Herremans, and Darrell Conklin (2016). "The Effect of Repetitive Structure on Enjoyment in Uplifting Trance Music". In: *International Conference on Music Perception and Cognition*. San-Francisco, United States. URL: https://hal.science/hal-01798672.

Agres, Kat, Dorien Herremans, Louis Bigo, and Darrell Conklin (2017). "Harmonic Structure Predicts the Enjoyment of Uplifting Trance Music". In: *Frontiers in Psychology* 7. DOI: 10.3389/fpsyg.2016.01999. URL: https://hal.science/hal-01798668.

Allegraud, Pierre, Louis Bigo, Laurent Feisthauer, Mathieu Giraud, Richard Groult, Emmanuel Leguy, and Florence Levé (2019). "Learning Sonata Form Structure on Mozart's String Quartets". In: *Transactions of the International Society for Music Information Retrieval (TISMIR)* 2.1, pp. 82–96. DOI: 10.5334/tismir.27. URL: https://hal.archives-ouvertes.fr/hal-02366640.

Bigo, Louis (2013). "Représentations symboliques musicales et calcul spatial". Theses. Université Paris-Est. URL: https://theses.hal.science/tel-01326827.

Bigo, Louis and Moreno Andreatta (2014). "A Geometrical Model for the Analysis of Pop Music ". In: *Sonus*. Sonus 35.1, pp. 36–48. URL: https://hal.archives-ouvertes.fr/hal-01263324.

— (2015). "Topological Structures in Computer-Aided Music Analysis". In: *Computational Music Analysis*. Ed. by David Meredith. Springer, pp. 57–80. DOI: 10.1007/978-3-319-25931-4_3. URL: https://hal.science/hal-01263349.

— (2017). "Towards Structural (Popular) Music Information Research". In: *European Music Analysis Conference (EuroMAC 2017)*. Strasbourg, France. URL: https://hal.archives-ouvertes.fr/hal-01517513.

— (2019). "Filtration of Pitch-Class Sets Complexes". In: *7th International Conference, MCM 2019*. Montiel M., Gomez-Martin F., Agustín-Aquino O. (eds) Mathematics and Computation in Music. MCM 2019. Lecture Notes in Computer Science, vol 11502. Springer, Cham. Madrid, Spain, pp. 213–226. DOI: 10.1007/978-3-030-21392-3_17. URL: https://hal.archives-ouvertes.fr/hal-02153236.

Bigo, Louis and Darrell Conklin (2015). "A viewpoint approach to symbolic music transformation". In: *11th International Symposium on Computer Music Multidisciplinary Research (CMMR) 2015*. Plymouth, United Kingdom. URL: https://hal.science/hal-01798678.

— (2016). *Four-part chorale transformation by harmonic template voicing*. Research Report. Universidad del País Vasco. URL: https://hal.science/hal-01802235.

Bigo, Louis, Jean-Louis Giavitto, and Antoine Spicher (2011). "Building Topological Spaces for Musical Objects". In: *Mathematics and Computation in Music*. cote interne IRCAM: Bigo11c. Paris, France, pp. 1–1. URL: https://hal.archives-ouvertes.fr/hal-01161290.

— (2013). "Spatial Programming for Musical Transformations and Harmonization". In: *Spatial Computing Workshiop (SCW)*. AAMAS satellite workshop W09. Saint-Paul, Minesota, United States, p. 9–16. URL: https://hal.science/hal-00925767.

Bigo, Louis and Antoine Spicher (2014). "Self-Assembly of Musical Representations in MGS". In: *International Journal of Unconventional Computing* 10.3, pp. 219–236. URL: https://hal.science/hal-02903099.

Bigo, Louis, Antoine Spicher, and Olivier Michel (2011a). "DIFFÉRENTES UTILISATIONS DE L'ESPACE EN MUSIQUE À L'AIDE D'UN LANGAGE DE PROGRAMMATION DÉDIÉ AU CALCUL SPATIAL". In: *Journées d'Informatique Musicale*. Saint-Etienne, France. URL: https://hal.science/hal-03104801.

Bigo, Louis, Antoine Spicher, and Olivier J.J. Michel (2010). "Spatial Programming for Music Representation and Analysis". In: *Spatial Computing Workshop 2010*. cote interne IRCAM: Bigo10b. Budapest, Hungary, pp. 1–1. URL: https://hal.archives-ouvertes.fr/hal-01161288.

— (2011b). "Différentes utilisations de l'espace en musique à l'aide d'un langage de programmation dédié au calcul spatial". In: *Journées d'Informatique Musicale 2011*. cote interne IRCAM: Bigo11a. St-Etienne, France, pp. 1–1. URL: https://hal.archives-ouvertes.fr/hal-01157093.

Bigo, Louis, Jérémie Garcia, Antoine Spicher, and Wendy E. Mackay (2012). "PaperTonnetz: Music Composition with Interactive Paper". In: *Sound and Music Computing*. Copenhague, Denmark. URL: https://hal.inria.fr/hal-00718334.

Bigo, Louis, Jean-Louis Giavitto, Moreno Andreatta, Olivier Michel, and Antoine Spicher (2013). "Computation and Visualization of Musical Structures in Chord-Based Simplicial Complexes". In: *MCM 2013 - 4th International Conference Mathematics and Computation in Music*. Ed. by Jason Yust, Jonathan Wild, and John Ashley Burgoyne. Vol. 7937. Lecture notes in computer science. Montreal, Canada: Springer, pp. 38–51. DOI: 10.1007/978-3-642-39357-0_3. URL: https://hal.archives-ouvertes.fr/hal-00925748.

Bigo, Louis, Antoine Spicher, Daniele Ghisi, and Moreno Andreatta (2014). "Spatial Transformations in Simplicial Chord Spaces". In: *Proceedings ICMC|SMC|2014*. cote interne IRCAM: Bigo14b. Athens, Greece, pp. 1112–1119. URL: https://hal.archives-ouvertes.fr/hal-01161081.

Bigo, Louis, Daniele Ghisi, Antoine Spicher, and Moreno Andreatta (2015). "Representation of Musical Structures and Processes in Simplicial Chord Spaces". In: *Computer Music Journal* 39.3, pp. 9 –24. DOI: 10.1162/COMJ_a_00312. URL: https://hal.archives-ouvertes.fr/hal-01263299.

Bigo, Louis, Mathieu Giraud, Richard Groult, Nicolas Guiomard-Kagan, and Florence Levé (2017). "Sketching Sonata Form Structure in Selected Classical String Quartets". In: *ISMIR 2017 - International Society for Music Information Retrieval Conference*. Suzhou, China. URL: https://hal.archives-ouvertes.fr/hal-01568703.

Bigo, Louis, Laurent Feisthauer, Mathieu Giraud, and Florence Levé (2018). "Relevance of musical features for cadence detection". In: *International Society for Music*

Information Retrieval Conference (ISMIR 2018). Paris, France. URL: https://hal.archives-ouvertes.fr/hal-01801060.

Conklin, Darrell and Louis Bigo (2015). "Trance generation by transformation". In: *8th International Workshop on Machine Learning and Music*. Vancouver, Canada. URL: https://hal.science/hal-01798675.

Cournut, Jules, Louis Bigo, Mathieu Giraud, and Nicolas Martin (2020). "Encodages de tablatures pour l'analyse de musique pour guitare". In: *Journées d'Informatique Musicale (JIM 2020)*. Strasbourg (en ligne), France. URL: https://hal.archives-ouvertes.fr/hal-02934382.

Cournut, Jules, Louis Bigo, Mathieu Giraud, Nicolas Martin, and David Régnier (2021). "What are the most used guitar positions?" In: *International Conference on Digital Libraries for Musicology (DLfM 2021)*. Online, United Kingdom, pp. 84–92. DOI: 10.1145/3469013.3469024. URL: https://hal.archives-ouvertes.fr/hal-03279863.

Couturier, Louis, Louis Bigo, and Florence Levé (2022a). "A dataset of symbolic texture annotations in Mozart piano sonatas". In: *International Society for Music Information Retrieval Conference (ISMIR 2022)*. Bengaluru, India.

— (2022b). "Annotating Symbolic Texture in Piano Music: a Formal Syntax". In: *Sound and Music Computing*. Saint-Etienne, France. URL: https://hal.archives-ouvertes.fr/hal-03631151.

Feisthauer, Laurent, Louis Bigo, and Mathieu Giraud (2019). "Modeling and learning structural breaks in sonata forms". In: *International Society for Music Information Retrieval Conference (ISMIR 2019)*. Delft, Netherlands. URL: https://hal.archives-ouvertes.fr/hal-02162936.

Feisthauer, Laurent, Louis Bigo, Mathieu Giraud, and Florence Levé (2020). "Estimating keys and modulations in musical pieces". In: *Sound and Music Computing Conference (SMC 2020)*. Simone Spagnol and Andrea Valle. Torino, Italy. URL: https://hal.archives-ouvertes.fr/hal-02886399.

Garcia, Jérémie, Louis Bigo, Antoine Spicher, and Wendy E. Mackay (2013). "PaperTonnetz: Supporting Music Composition with Interactive Paper". In: *Extend Abstract on Human Factors in Computing Systems*. Paris, France. URL: https://hal.inria.fr/hal-00837640.

Karystinaios, Emmanouil, Corentin Guichaoua, Moreno Andreatta, Louis Bigo, and Isabelle Bloch (2020). "Musical genre descriptor for classification based on Tonnetz trajectories". In: *Journées d'Informatique Musicale*. Strasbourg, France. URL: https://hal.science/hal-03031287.

Keller, Mikaela, Gabriel Loiseau, and Louis Bigo (2021). "What Musical Knowledge Does Self-Attention Learn?" In: *Workshop on NLP for Music and Spoken Audio (NLP4MuSA 2021)*. Online, France. URL: https://hal.archives-ouvertes.fr/hal-03419236.

Keller, Mikaela, Kamil Akesbi, Lorenzo Moreira, and Louis Bigo (2021). "Techniques de traitement automatique du langage naturel appliquées aux représentations symboliques musicales". In: *JIM 2021 - Journées d'Informatique Musicale*. Virtual, France. URL: https://hal.archives-ouvertes.fr/hal-03279850.

Kermarec, Mathieu, Louis Bigo, and Mikaela Keller (2022). *Improving Tokenization Expressiveness With Pitch Intervals*. 23rd International Society for Music Information Retrieval Conference (ISMIR 2022), Late-Breaking Demo Session. Poster. URL: https://hal.archives-ouvertes.fr/hal-03877642.

Micchi, Gianluca, Louis Bigo, Mathieu Giraud, Richard Groult, and Florence Levé (2021). "I Keep Counting: An Experiment in Human/AI Co-creative Songwriting". In: *Transactions of the International Society for Music Information Retrieval (TISMIR)*.

Régnier, David, Nicolas Martin, and Louis Bigo (2021). "Identification of rhythm guitar sections in symbolic tablatures". In: *International Society for Music Information Retrieval Conference (ISMIR 2021)*. Online, United States. URL: https://hal.archives-ouvertes.fr/hal-03335822.

Bibliography

Agostini, Andrea and Daniele Ghisi (2015). "A max library for musical notation and computer-aided composition". In: *Computer Music Journal* 39.2, pp. 11–27.

Ariga, Shunya, Satoru Fukayama, and Masataka Goto (2017). "Song2Guitar: A Difficulty-Aware Arrangement System for Generating Guitar Solo Covers from Polyphonic Audio of Popular Music." In: *International Society for Music Information Retrieval Conference (ISMIR 2017)*, pp. 568–574.

Assayag, Gérard and Shlomo Dubnov (2004). "Using Factor Oracles for Machine Improvisation". In: *Soft Computing* 8.9, pp. 604–610. DOI: 10.1007/s00500-004-0385-4.

Barbancho, Ana M, Anssi Klapuri, Lorenzo J Tardón, and Isabel Barbancho (2011). "Automatic transcription of guitar chords and fingering from audio". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.3, pp. 915–921.

Bell, Adam Patrick (2018). *Dawn of the DAW: The studio as musical instrument*. Oxford University Press.

Ben-Tal, Oded, Matthew Tobias Harris, and Bob LT Sturm (2021). "How Music AI Is Useful: Engagements with Composers, Performers and Audiences". In: *Leonardo* 54.5, pp. 510–516.

Bimbot, Frédéric, Emmanuel Deruty, Gabriel Sargent, and Emmanuel Vincent (2016). "System & contrast: a polymorphous model of the inner organization of structural segments within music pieces". In: *Music Perception: An Interdisciplinary Journal* 33.5, pp. 631–661.

Blombach, Ann (1987). "Phrase and Cadence: A Study of Terminology and Definition." In: *Journal of Music Theory Pedagogy* 1, 225–51.

Bogdanov, Dmitry et al. (2013). "Essentia: An audio analysis library for music information retrieval". In: *International Society for Music Information Retrieval Conference (ISMIR 2013)*.

Bresson, Jean, Carlos Agon, and Gérard Assayag (2011). "OpenMusic: visual programming environment for music composition, analysis and research". In: *Proceedings of the 19th ACM international conference on Multimedia*, pp. 743–746.

Briot, Jean-Pierre, Gaëtan Hadjeres, and François-David Pachet (2019). *Deep learning techniques for music generation*. Springer. ISBN: 978-3-319-70162-2.

Briot, Jean-Pierre and François Pachet (2020). "Deep learning for music generation: challenges and directions". In: *Neural Computing and Applications* 32.4, pp. 981–993.

Brooks, Frederick P, AL Hopkins, Peter G Neumann, and William V Wright (1957). "An experiment in musical composition". In: *IRE Transactions on Electronic Computers* 3, pp. 175–182.

Burns, Anne-Marie and Marcelo M Wanderley (2006). "Visual methods for the retrieval of guitarist fingering". In: *International Conference on New Interfaces for Musical Expression (NIME 2006)*. Citeseer, pp. 196–199.

Callender, Clifton, Ian Quinn, and Dmitri Tymoczko (2008). "Generalized voice-leading spaces". In: *Science* 320.5874, pp. 346–348.

Calvo-Zaragoza, Jorge, Jan Hajič Jr, and Alexander Pacha (2020). "Understanding optical music recognition". In: *ACM Computing Surveys (CSUR)* 53.4, pp. 1–35.

Cancino-Chacón, Carlos Eduardo et al. (2022). "Partitura: A Python Package for Symbolic Music Processing". In: *Proceedings of the Music Encoding Conference (MEC2022)*. Halifax, Canada.

Caplin, William E (2001). *Classical form: A theory of formal functions for the instrumental music of Haydn, Mozart, and Beethoven*. Oxford University Press.

Caplin, William E. (2004). "The Classical Cadence: Conceptions and Misconceptions." In: *Journal of the American Musicological Society* 57, pp. 51–117.

Chen, Tsung-Ping and Li Su (2021). "Attend to chords: Improving harmonic analysis of symbolic music using transformer-based models". In: *Transactions of the International Society for Music Information Retrieval* 4.1.

Chen, Yu-Hua, Yu-Hsiang Huang, Wen-Yi Hsiao, and Yi-Hsuan Yang (2020). "Automatic composition of guitar tabs by transformers and groove modeling". In:

Chen, Yuan-Ping, Li Su, Yi-Hsuan Yang, et al. (2015). "Electric Guitar Playing Technique Detection in Real-World Recording Based on F0 Sequence Pattern Recognition." In: *International Society for Music Information Retrieval Conference (ISMIR 2015)*, pp. 708–714.

Chew, Elaine (2000). "Towards a mathematical model of tonality". PhD thesis. Massachusetts Institute of Technology.

Chou, Yi-Hui, I Chen, Chin-Jui Chang, Joann Ching, Yi-Hsuan Yang, et al. (2021). "MidiBERT-piano: large-scale pre-training for symbolic music understanding". In: *arXiv preprint arXiv:2107.05223*.

Chuan, Ching-Hua, Kat Agres, and Dorien Herremans (2020). "From context to concept: exploring semantic relationships in music with word2vec". In: *Neural Computing and Applications* 32.4, pp. 1023–1036.

Cífka, Ondřej, Umut Şimşekli, and Gaël Richard (2020). "Groove2Groove: one-shot music style transfer with supervision from synthetic data". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28, pp. 2638–2650.

Conklin, D. and J. G. Cleary (1988). "Modelling and Generating Music Using Multiple Viewpoints". In: *Proceedings of the First Workshop on AI and Music*. The American Association for Artificial Intelligence, pp. 125–137.

Conklin, Darrell (2013). "Multiple viewpoint systems for music classification". In: *Journal of New Music Research* 42.1, pp. 19–26.

Conklin, Darrell and Christina Anagnostopoulou (2001). "Representation and Discovery of Multiple Viewpoint Patterns". In: *International Computer Music Conference (ICMC 2001)*, pp. 479–485.

Conklin, Darrell and Ian H Witten (1995). "Multiple viewpoint systems for music prediction". In: *Journal of New Music Research* 24.1, pp. 51–73.

Cooke, D. (1990). *The Language of Music*. Clarendon paperbacks. Oxford University Press. ISBN: 9780198161806. URL: https://books.google.com.sg/books?id=92Zg1lO4tCEC.

Cunha, Nailson dos Santos, Anand Subramanian, and Dorien Herremans (2018). "Generating guitar solos by integer programming". In: *Journal of the Operational Research Society* 69.6, pp. 971–985.

Cuthbert, Michael Scott and Christopher Ariza (2010). "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data". In: *International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 637–642. DOI: 10.5281/zenodo.1416114.

Dahlig-Turek, Ewa, Sebastian Klotz, Richard Parncutt, and Frans Wiering, eds. (2012). *Musicology (Re-) Mapped*. Discussion Paper, European Science Foundation. URL: http://www.esf.org/publications.html.

Dai, Shuqi, Zheng Zhang, and Gus G Xia (2018). "Music style transfer: A position paper". In: *arXiv preprint arXiv:1803.06841*.

Das, Orchisama, Blair Kaneshiro, and Tom Collins (2018). "Analyzing and classifying guitarists from rock guitar solo tablature". In: *Sound and Music Computing Conference (SMC 2018)*.

Deruty, Emmanuel, Maarten Grachten, Stefan Lattner, Javier Nistal, and Cyran Aouameur (2022). "On the Development and Practice of AI Technology for Contemporary Popular Music Production". In: *Transactions of the International Society for Music Information Retrieval* 5.1.

Devaney, Johanna, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula (2015). "Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis". In: *International Society for Music Information Retrieval Conference (ISMIR 2015)*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT 2019)*. Association for Computational Linguistics, pp. 4171–4186.

Douwes, Constance, Philippe Esling, and Jean-Pierre Briot (2021). "Energy Consumption of Deep Generative Audio Models". In: *arXiv preprint arXiv:2107.02621*.

Esling, Philippe and Ninon Devis (2020). "Creativity in the era of artificial intelligence". In: *arXiv:2008.05959*.

Esling, Philippe, Naotake Masuda, Adrien Bardet, Romeo Despres, et al. (2019). "Universal audio synthesizer control with normalizing flows". In:

Euler, Leonhard (1739). "Tentamen novæ theoriæ musicæ ex certissismis harmoniæ principiis dilucide expositae". In: *Saint Petersburg Academy*.

Ferretti, Stefano (2016). "Guitar solos as networks". In: *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, pp. 1–6.

Fradet, Nathan, Jean-Pierre Briot, Fabien Chhel, Amal El Fallah-Seghrouchni, and Nicolas Gutowski (2021). "MidiTok: A Python Package for MIDI File Tokenization". In: *International Society for Music Information Retrieval Conference (ISMIR 2021), Late-Breaking Demo Session*. Online, United States.

Garczynski, Louis, Mathieu Giraud, Emmanuel Leguy, and Philippe Rigaux (2022). "Modeling and Editing Cross-Modal Synchronization on a Label Web Canvas". In: *Music Encoding Conference (MEC 2022)*.

Gaston, E Thayer (1958). "Functional music". In: *Teachers College Record* 59.9, pp. 292–309.

Giavitto, Jean-Louis and Olivier Michel (2003). "Modeling the topological organization of cellular processes". In: *BioSystems* 70.2, pp. 149–163.

Giavitto, Jean-Louis, Olivier Michel, Julien Cohen, and Antoine Spicher (2004). "Computations in space and space in computations". In: *International Workshop on Unconventional Programming Paradigms*. Springer, pp. 137–152.

Gioti, Artemi-Maria (2021). "Artificial intelligence for music composition". In: *Handbook of Artificial Intelligence for Music*. Springer, pp. 53–73.

Giraud, Mathieu, Richard Groult, Emmanuel Leguy, and Florence Levé (2015). "Computational Fugue Analysis". In: *Computer Music Journal* 39.2.

Grachten, Maarten, Stefan Lattner, and Emmanuel Deruty (2020). "Bassnet: A variational gated autoencoder for conditional generation of bass guitar tracks with learned interactive control". In: *Applied Sciences* 10.18, p. 6627.

Guo, Zixun, J. Kang, and D. Herremans (2023). "A Domain-Knowledge-Inspired Music Embedding Space and a Novel Attention Mechanism for Symbolic Music Modeling". In: *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. AAAI. Washington DC: AAAI.

Hadjeres, Gaëtan and Léopold Crestel (2021). "The piano inpainting application". In: *arXiv preprint arXiv:2107.05944*.

Hadjeres, Gaëtan, François Pachet, and Frank Nielsen (2017). "Deepbach: a steerable model for bach chorales generation". In: pp. 1362–1371.

Hawthorne, Curtis et al. (2019). "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset". In: *International Conference on Learning Representations (ICLR 2019)*.

Hentschel, Johannes, Markus Neuwirth, and Martin Rohrmeier (2021). "The annotated Mozart sonatas: Score, harmony, and cadence". In: *Transactions of the International Society for Music Information Retrieval* 4.1, pp. 67–80.

Hepokoski, James and Warren Darcy (1997). "The Medial Caesura and Its Role in the Eighteenth-Century Sonata Exposition". In: *Music Theory Spectrum* 19.2, pp. 115–154.

— (2006). *Elements of Sonata Theory: Norms, Types, and Deformations in the Late-Eighteenth-Century Sonata*. Oxford University Press. ISBN: 9780199773916.

Herremans, Dorien, Ching-Hua Chuan, and Elaine Chew (2017). "A functional taxonomy of music generation systems". In: *ACM Computing Surveys* 50.5, pp. 1–30.

Hiller, Lejaren Arthur and Leonard Maxwell Isaacson (1959). *Experimental music: composition with an electronic computer*. McGraw-Hill.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hori, Gen and Shigeki Sagayama (2016). "Minimax Viterbi Algorithm for HMM-Based Guitar Fingering Decision." In: *International Society for Music Information Retrieval Conference (ISMIR 2016)*, pp. 448–453.

Hsiao, Wen-Yi, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang (2021). "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 1, pp. 178–186.

Huang, Anna, Monica Dinculescu, Ashish Vaswani, and Doug Eck (2018). "Visualizing Music Self-Attention". In: *Conference on Neural Information Processing Systems (NIPS 2018)*.

Huang, Cheng-Zhi Anna et al. (2019). "Music Transformer: Generating Music with Long-term Structure". In: *International Conference on Learning Representations (ICLR 2019)*.

Huang, Cheng-Zhi Anna, Hendrik Vincent Koops, Ed Newton-Rex, Monica Dinculescu, and Carrie J. Cai (2020). "AI Song Contest: Human-AI Co-Creation in Songwriting". In: *International Society for Music Information Retrieval Conference (ISMIR 2020)*.

Huang, Yu-Siang and Yi-Hsuan Yang (2020). "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions". In: *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1180–1188.

Huron, David (2008). *Sweet anticipation: Music and the psychology of expectation*. MIT press.

Jackendoff, Ray (2009). "Parallels and nonparallels between language and music". In: *Music perception* 26.3, pp. 195–204.

Jackendoff, Ray and Fred Lerdahl (2006). "The capacity for music: What is it, and what's special about it?" In: *Cognition* 100.1, pp. 33–72.

Jurafsky, Dan and James H Martin (2014). "Speech and language processing. Vol. 3". In: *US: Prentice Hall*.

Karystinaios, Emmanouil and Gerhard Widmer (2022). "Cadence Detection in Symbolic Classical Music using Graph Neural Networks". In: *International Society for Music Information Retrieval Conference (ISMIR 2022)*.

Kawai, Lisa, Philippe Esling, and Tatsuya Harada (2020). "Attributes-Aware Deep Music Transformation." In: *International Society for Music Information Retrieval Conference (ISMIR 2020)*, pp. 670–677.

Kayacik, Claire et al. (2019). "Identifying the intersections: User experience+ research scientist collaboration in a generative machine learning interface". In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–8.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Kong, Qiuqiang, Bochen Li, Jitong Chen, and Yuxuan Wang (2022). "Giantmidi-piano: A large-scale midi dataset for classical piano music". In: *Transactions of the International Society for Music Information Retrieval* 5.1, pp. 87–98.

Lähdeoja, Otso, Benoît Navarret, Santiago Quintans, and Anne Sedes (2010). "The Electric Guitar: An Augmented Instrument and a Tool for Musical Composition." In: *Journal of interdisciplinary music studies* 4.2.

Lerdahl, Fred and Ray Jackendoff (1983). *A Generative Theory of Tonal Music*. MIT Press.

Liang, Beici, György Fazekas, and Mark Sandler (2019). "Transfer learning for piano sustain-pedal detection". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–6.

Liu, Jiafeng et al. (2022). "Symphony Generation with Permutation Invariant Language Model". In: *International Society for Music Information Retrieval Conference (ISMIR 2022)*.

Liutkus, Antoine et al. (2021). "Relative positional encoding for transformers with linear complexity". In: *International Conference on Machine Learning*. PMLR, pp. 7067–7079.

Low, Thomas et al. (2019). "The ISMIR Explorer-A Visual Interface for Exploring 20 Years of ISMIR Publications." In: *International Society for Music Information Retrieval Conference (ISMIR 2019)*, pp. 754–760.

Lupker, Jeffrey AT (2021). "Score-Transformer: A Deep Learning Aid for Music Composition". In: *International Conference on New Interfaces for Musical Expression (NIME 2021)*. PubPub.

Madjiheurem, Sephora, Lizhen Qu, and Christian Walder (2016). "Chord2vec: Learning musical chord embeddings". In: *Proceedings of the constructive machine learning workshop at 30th conference on neural information processing systems (NIPS2016), Barcelona, Spain*.

Makris, D., K. Agres, and D. Herremans (2021). "Generating Lead Sheets with Affect: A Novel Conditional seq2seq Framework". In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE. Virtual: IEEE. URL: https://arxiv.org/abs/2104.13056.

Mazzola, Guerino (2012). *The topos of music: geometric logic of concepts, theory, and performance*. Birkhäuser.

McFee, Brian et al. (2015). "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8, pp. 18–25.

McVicar, Matt, Satoru Fukayama, and Masataka Goto (2014a). "AutoLeadGuitar: Automatic generation of guitar solo phrases in the tablature space". In: *2014 12th International Conference on Signal Processing (ICSP)*. IEEE, pp. 599–604.

— (2014b). "Autorhythmguitar: Computer-aided composition for rhythm guitar in the tab space". In: *International Computer Music Conference (ICMC 2014)*.

— (2015). "AutoGuitarTab: Computer-aided composition of rhythm and lead guitar parts in the tablature space". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.7, pp. 1105–1117.

Micchi, Gianluca, Mark Gotham, and Mathieu Giraud (2020). "Not All Roads Lead to Rome: Pitch Representation and Model Architecture for Automatic Harmonic Analysis". In: *Transactions of the International Society for Music Information Retrieval (TISMIR)* 3.1, pp. 42–54. DOI: 10.5334/tismir.45. URL: https://hal.archives-ouvertes.fr/hal-02934374.

Mielke, Sabrina J et al. (2021). "Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP". In: *arXiv preprint arXiv:2112.10508*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26.

Müller, Meinard (2015). *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Vol. 5. Springer.

Muller, Meinard, Daniel PW Ellis, Anssi Klapuri, and Gaël Richard (2011). "Signal processing for music analysis". In: *IEEE Journal of selected topics in signal processing* 5.6, pp. 1088–1110.

Navarret, Benoît (2013). "Caractériser la guitare électrique: définitions, organologie et analyse de données verbales". PhD thesis. Paris 8.

Neuwirth, Markus, Daniel Harasim, Fabian C. Moss, and Martin Rohrmeier (2018). "The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets". In: *Frontiers in Digital Humanities* 5, p. 16. ISSN: 2297-2668. DOI: 10.3389/fdigh.2018.00016. URL: https://www.frontiersin.org/article/10.3389/fdigh.2018.00016.

Normand, Quentin (2021). "Modélisation de techniques de jeu dans les tablatures de guitare". MA thesis. Université de Lille, Polytech'Lille.

Oore, Sageev, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan (2020). "This time with feeling: Learning expressive musical performance". In: *Neural Computing and Applications* 32.4, pp. 955–967.

Pearce, Marcus Thomas (2005). "The construction and evaluation of statistical models of melodic structure in music perception and composition". PhD thesis. City University London.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training". In: OpenAI.

Reboursière, Loïc et al. (2012). "Left and right-hand guitar playing techniques detection." In: *International Conference on New Interfaces for Musical Expression (NIME 2012)*.

Reif, Emily et al. (2019). "Visualizing and Measuring the Geometry of BERT". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 8592–8600.

Roberts, Adam, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck (2018). "A hierarchical latent vector model for learning long-term structure in music". In: *International Conference on Machine Learning*. PMLR, pp. 4364–4373.

Roberts, Adam et al. (2019). "Magenta studio: Augmenting creativity with deep learning in ableton live". In: *International Workshop on Musical Metacreation (MUME 2019)*.

Rosen, Charles (1997). *The Classical Style: Haydn, Mozart, Beethoven*. 653. WW Norton & Company.

Sarmento, Pedro et al. (2021). "DadaGP: A dataset of tokenized GuitarPro songs for sequence models". In: *International Society for Music Information Retrieval Conference (ISMIR 2021)*.

Schuster, Mike and Kaisuke Nakajima (2012). "Japanese and korean voice search". In: *IEEE international conference on acoustics, speech and signal processing (ICASSP 2012)*. IEEE, pp. 5149–5152.

Sears, David R. W., William E. Caplin, and Stephen McAdams (2014). "Perceiving the Classical Cadence". In: *Music Perception. An Interdisciplinary Journal* 31.5, pp. 397–417. ISSN: 0730-7829. DOI: 10.1525/mp.2014.31.5.397. eprint: http://mp.ucpress.edu/content/31/5/397.full.pdf. URL: http://mp.ucpress.edu/content/31/5/397.

Sears, David R. W., Marcus T. Pearce, William E. Caplin, and Stephen McAdams (2017). "Simulating melodic and harmonic expectations for tonal cadences using probabilistic models". In: *Journal of New Music Research* 47.1, pp. 29–52. DOI: 10.1080/09298215.2017.1367010. eprint: https://doi.org/10.1080/09298215.2017.1367010. URL: https://doi.org/10.1080/09298215.2017.1367010.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2015). "Neural machine translation of rare words with subword units". In: *arXiv preprint arXiv:1508.07909*.

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani (2018). "Self-Attention with Relative Position Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 464–468.

Tenkanen, Atte and Fernanda Gualda (2008). "Detecting changes in musical texture". In: *International Workshop on Machine Learning and Music*.

Trades, Music, ed. (2021). *The Music Industry Census*.

Tymoczko, Dmitri (2010). *A geometry of music: Harmony and counterpoint in the extended common practice*. Oxford University Press.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems (NIPS 2017)* 30.

Wang, Ziyu and Gus Xia (2021). "MuseBERT: Pre-training Music Representation for Music Understanding and Controllable Generation." In: *International Society for Music Information Retrieval Conference (ISMIR 2021)*, pp. 722–729.

Wang, Ziyu, Dingsu Wang, Yixiao Zhang, and Gus Xia (2020). "Learning interpretable representation for controllable polyphonic music generation". In: *International Society for Music Information Retrieval Conference (ISMIR 2020)*.

Wiggins, Andrew and Youngmoo Kim (2019). "Guitar Tablature Estimation with a Convolutional Neural Network." In: *International Society for Music Information Retrieval Conference (ISMIR 2019)*, pp. 284–291.

Xi, Qingyang, Rachel M Bittner, Johan Pauwels, Xuzhou Ye, and Juan Pablo Bello (2018). "GuitarSet: A Dataset for Guitar Transcription." In: *International Society for Music Information Retrieval Conference (ISMIR 2018)*, pp. 453–460.

Yang, Li-Chia and Alexander Lerch (2020). "On the evaluation of generative models in music". In: *Neural Computing and Applications* 32.9, pp. 4773–4784.

Yazawa, Kazuki, Katsutoshi Itoyama, and Hiroshi G Okuno (2014). "Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3122–3126.

Zeng, Mingliang et al. (2021). "Musicbert: Symbolic music understanding with large-scale pre-training". In: *Findings of the Association for Computational Linguistics (ACL-IJCNLP 2021)*.