
**Institut National des Langues et Civilisations
Orientales**

Département Textes, Informatique, Multilinguisme

Titre du mémoire

MASTER
TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Martin DIGARD

Directeur de mémoire :

Damien NOUVEL

Encadrant :

Florent JACQUEMARD

Année universitaire 2020-2021

TABLE DES MATIÈRES

Liste des figures	5
Liste des tableaux	5
Introduction générale	7
1 Contexte	9
1.1 TAL et MIR	9
1.2 La transcription automatique de la musique	11
1.3 La transcription automatique de la batterie	14
1.4 Les représentations de la musique	14
2 État de l'art	19
2.1 Monophonique et polyphonique	19
2.2 Audio vers MIDI	20
2.3 MIDI vers partition	21
2.4 Approche linéaire et approche hiérarchique	21
3 Méthodes	25
3.1 La notation de la batterie	25
3.2 Modélisation pour la transcription	31
3.3 Qparse	32
3.4 Les systèmes	33
4 Expérimentations	39
4.1 Corpus	39
4.2 Analyse MIDI-Audio	41
4.3 Expérimentation théorique d'un système	46
4.4 Développement	50
5 Discussion	55
5.1 Travaux réalisés	55
5.2 Travaux non-réalisés	55
5.3 Travaux futures	56
Conclusion générale	57

Bibliographie**59**

LISTE DES FIGURES

1.1	Transcription automatique	13
1.2	Exemple évènements avec durée	15
1.3	Critère pour un évènement	16
1.4	Exemple évènements sans durée	16
1.5	Exemple de partition de piano	16
1.6	MusicXML	17
2.1	HMM	22
2.2	arbre_jazz	23
3.1	Rapport des figures de notes	26
3.2	Hauteur et têtes de notes	27
3.3	Point et liaison	28
3.4	Les silences	28
3.5	Silence joué	29
3.6	Équivalence	29
3.7	Nuances	30
3.8	Séparation des voix	30
4.1	référence 1	46
4.2	Motifs	47

LISTE DES TABLEAUX

3.1	Pitches et instruments	31
3.2	Vélocité et nuances	31
3.3	Systèmes	35

INTRODUCTION GÉNÉRALE

Ce mémoire de recherche, effectué en parallèle d'un stage à l'Inria dans le cadre du master de traitement automatique des langues de l'Inalco, contient une proposition originale ainsi que diverses contributions ayant toutes pour objectif d'améliorer **qparselib**, un outil de transcription automatique de la musique sur sa capacité à transcrire la batterie. Nous ne parlerons donc pas directement de langues naturelles, mais de l'écriture automatique de partitions de musique à partir de données audios ou symboliques. Cette exercice nécessitera la manipulation d'un langage musical codifié avec une grammaire (solfège, durées, nuances, volumes) et soulèvera des problématiques concernées par les techniques du traitement automatique des langues.

L'écriture musicale offre de nombreuses possibilités pour la transcription d'un rythme donné. Le contexte musical ainsi que la lisibilité d'une partition pour un batteur entraîné conditionnent les choix d'écritures. Reconnaître la métrique principale d'un rythme, la façon de regrouper les notes par les ligatures, ou simplement décider d'un usage pour une durée parmi les différentes continuations possibles (notes pointées, liaisons, silences, etc.) constituent autant de possibilités que de difficultés.

Voici la proposition de ce mémoire ainsi que les contributions apportées lors du stage :

- Proposition principale : les systèmes (3.4) :
Recherche de rythmes génériques en amont dans la chaîne de traitement.
⇒ L'objectif de fixer des choix le plus tôt possible afin de simplifier le reste des calculs en éliminant une partie d'entre eux. Ces choix concernent notamment la métrique et les règles de réécriture.
- Une description de la notation de la batterie (??)
- Une modélisation de la transcription de la batterie (3.2)
- Analyse MIDI-Audio (4.2)
- Théorie et tests unitaires pour le passage au polyphonique (4.4)
- Création de grammaires pondérées pour la batterie (4.4)
- Contributions sur la branche « distance » dans :
 - `qparselib/notes/cluster.md`
 - `qparselib/src/segment/import/` :
DrumCode hpp et cpp

Nous présenterons le contexte suivi d'un état de l'art et nous définirons de manière générale le processus de transcription automatique de la musique pour enfin étayer les méthodes utilisées pour la transcription automatique de la batterie, et nous présenterons les principales contributions apportées à l'outil qparse. Nous décrirons ensuite le corpus ainsi que les différentes expérimentations menées. Nous concluerons par une discussion sur les résultats obtenus et les pistes d'améliorations futures à explorer.

CONTEXTE

Sommaire

1.1	TAL et MIR	9
1.2	La transcription automatique de la musique	11
1.3	La transcription automatique de la batterie	14
1.4	Les représentations de la musique	14
1.4.1	Le format MusicXML	17

Introduction

La transcription automatique de la musique (AMT) est un défi ancien [1] et difficile qui n'est toujours pas résolu. Il a engendré une pluie de sous-tâches qui ont donné naissance au domaine de la recherche d'information musicale (MIR). Actuellement, de nombreux travaux de MIR font appel au traitement automatique des langues (TAL)¹.

Dans ce chapitre, nous parlerons de l'informatique musicale, nous tenterons d'établir les liens existants entre le MIR et le TAL ainsi qu'entre les notions de langage musical et langue naturelle. Nous traiterons également de l'utilité et du problème de l'AMT et de la transcription automatique de la batterie (ADT).

Enfin, nous décrirons les représentations de la musique qui sont nécessaires à la compréhension du présent travail.

1.1 TAL et MIR

L'informatique musicale² est une étude du traitement de la musique [2], en particulier des représentations musicales, de la transformée de

1. NLP4MuSA, the 2nd Workshop on Natural Language Processing for Music and Spoken Audio, co-located with ISMIR 2021.

2. https://en.wikipedia.org/wiki/Music_informatics

Fourier pour la musique³, de l'analyse de la structure de la musique et de la reconnaissance des accords. D'autres sujets de recherche en informatique musicale comprennent la modélisation informatique de la musique, l'analyse informatique de la musique, la reconnaissance optique de la musique, les éditeurs audio numériques, les moteurs de recherche de musique en ligne, la recherche d'informations musicales et les questions cognitives dans la musique.

Le MIR^{4 5} apparaît vers le début des années 2000 [3]. C'est une science interdisciplinaire qui fait appel à de nombreux domaines comme la musicologie, l'analyse musicale, la psychologie, les sciences de l'information, le traitement du signal et les méthodes d'apprentissage automatisé en informatique. Cette discipline récente a notamment été soutenue par de grandes compagnies du web qui veulent développer des systèmes de recommandation de musique ou des moteurs de recherche dédiés au son et à la musique.

Is Music a Language?



Leonard Bernstein

Norton Lectures at Harvard, 1973

« The Unanswered Question: Six Talks at Harvard »

idea of music as a kind of universal language

notion of a worldwide, « inborn musical grammar »

cf. **Noam Chomsky** « Language and Mind »

theory of innate grammatical competence

Aborder la musique à travers le TAL nécessite une réflexion autour de la musique en tant que langage ainsi que la possibilité de comparer ce même langage avec les langues naturelles. Quelques travaux en neurosciences ont abordé la question, notamment par observation des

3. <https://interstices.info/de-fourier-a-la-reconnaissance-musicale/>

4. <https://ismir.net/>

5. <https://ismir2021.ismir.net/>

processus cognitifs et neuronaux que les systèmes de traitement de ces deux langages avaient en commun. Dans le travail de Poulin-Charronnat *et al.* [4], la musique est reconnue comme étant un système complexe spécifique à l'être humain dont une des similitudes avec les langues naturelles est l'émergence de régularités reconnues implicitement par le système cognitif. La question de la pertinence de l'analogie entre langues naturelles et langage musical a également été soulevée à l'occasion de projets de recherche en TAL. Keller *et al.* [5] ont exploré le potentiel de ces techniques à travers les plongements de mots et le mécanisme d'attention pour la modélisation de données musicales. La question du sens d'une phrase musicale apparaît, selon eux, à la fois comme une limite et un défi majeur pour l'étude de cette analogie.

Ici, Digression sur la musicologie calculatoire vs linguistique computationnelle ?

D'autres travaux très récents, ont aussi été révélés lors de la *première conférence sur le NLP pour la musique et l'audio (NLP4MusA 2020)*. Lors de cette conférence, Jiang *et al.* [6] ont présenté leur implémentation d'un modèle de langage musical auto-attentif visant à améliorer le mécanisme d'attention par élément, déjà très largement utilisé dans les modèles de séquence modernes pour le texte et la musique.

Il semblerait que le domaine du TAL qui se rapproche le plus du MIR soit la reconnaissance de la parole. En effet, la séparation des sources ont des approches similaires dans les deux domaines. De plus, il existe un lien entre partition musicale comme manière d'écrire la musique et texte comme manière d'écrire la parole.

Similitudes :

Reconnaissance automatique de la parole :

signal \Rightarrow phonèmes \Rightarrow texte Transcription automatique de la musique :

signal \Rightarrow MIDI \Rightarrow partition Différence :

Texte (données linéaires) \neq partition (données structurées hiérarchiques)

1.2 La transcription automatique de la musique

En musique, la transcription⁶ est la pratique consistant à noter un morceau ou un son qui n'était auparavant pas noté et/ou pas populaire en tant que musique écrite, par exemple, une improvisation de jazz ou une bande sonore de jeu vidéo. Lorsqu'un musicien est chargé de créer une partition à partir d'un enregistrement et qu'il écrit les notes qui composent le morceau en notation musicale, on dit qu'il a créé une transcription musicale de cet enregistrement.

6. [https://en.wikipedia.org/wiki/Transcription_\(music\)](https://en.wikipedia.org/wiki/Transcription_(music))

L'objectif de la transcription automatique de la musique (AMT) [7] est de convertir la performance d'un musicien en notation musicale - un peu comme la conversion de la parole en texte dans le traitement du langage naturel. L'AMT a des intérêts multiples, notamment pour la transcription de solos ou encore pour la constitution de corpus musicologiques, ou encore pour l'interprétation de la musique et l'analyse du contenu musical [8]. Par exemple, un grand nombre de fichiers audio et vidéo musicaux sont disponibles sur le Web, et pour la plupart d'entre eux, il est difficile de trouver les partitions musicales correspondantes, qui sont nécessaires pour pratiquer la musique, faire des reprises et effectuer une analyse musicale détaillée. Les partitions de musique classique sont facilement accessibles et il y a peu de demandes de nouvelles transcriptions. D'un point de vue pratique, des demandes beaucoup plus commerciales et académiques sont attendues dans le domaine de la musique populaire [8]. Les modèles grammaticaux qui représentent la structure hiérarchique des séquences d'accords se sont avérés très utiles dans les analyses récentes de l'harmonie du jazz [9]. Comme déjà évoqué précédemment, il s'agit d'un problème ancien et difficile. C'est un « graal » de l'informatique musicale. En 1976, H. C. Longuet-Higgins [1] évoquait déjà la représentation musicale en arbre syntaxique dans le but d'écrire automatiquement des partitions à partir de données audio en se basant sur un mimétisme psychologique de l'approche humaine. De même pour les chercheurs en audio James A. Moorer, Martin Piszczalski et Bernard Galler qui, en 1977⁷, ont utilisé leurs connaissances en ingénierie de l'audio et du numérique pour programmer un ordinateur afin de lui faire analyser un enregistrement musical numérique de manière à détecter les lignes mélodiques, les accords et les accents rythmiques des instruments à percussion.

La tâche de transcription automatique de la musique comprend deux activités distinctes : l'analyse d'un morceau de musique et l'impression d'une partition à partir de cette analyse.

La figure 1.1 est une proposition de Benetos et al. (mettre le « et al. » en italique?) [7] qui représente l'architecture générale d'un système de transcription musicale. On y observe plusieurs sous-tâches de l'AMT :

- La séparation des sources à partir de l'audio.
- Le système de transcription :
 - Cœur du système :
 - ⇒ Algorithmes de détection des multi-pitches et de suivi des notes.
 - Quatres sous-tâches optionnelles accompagnent ces algorithmes :

7. [https://en.wikipedia.org/wiki/Transcription_\(music\)](https://en.wikipedia.org/wiki/Transcription_(music))

- identification de l'instrument;
- estimation de la tonalité et de l'accord;
- détection de l'apparition et du décalage;
- estimation du tempo et du rythme.
- Apprentissage sur des modèles acoustiques et musicologiques.
- *Optionnel* : Informations fournies de manière externe, soit fournie en amont (genre, instruments, . . .), soit par interaction avec un utilisateur (infos sur une partition incomplète).



FIGURE 1.1 – Transcription automatique

Les sous-systèmes et algorithmes optionnels sont présentés à l'aide de lignes pointillées. Les doubles flèches mettent en évidence les connexions entre les systèmes qui incluent la fusion d'informations et une communication plus interactive entre les systèmes.

1.3 La transcription automatique de la batterie

La batterie est un instrument récent qui s'est longtemps passé de partition. En effet pour un batteur, la qualité de lecteur lorsqu'elle était nécessaire, résidait essentiellement dans sa capacité à lire les partitions des autres instrumentistes (par exemple, les grilles d'accords et la mélodie du thème en jazz) afin d'improviser un accompagnement approprié que personne ne pouvait écrire pour lui à sa place.

Les partitions de batterie sont arrivées par nécessité avec la pédagogie et l'émergence d'écoles de batterie partout dans le monde. Un autre facteur qui a contribué à l'expansion des partitions de batterie est l'émergence de la musique assistée par ordinateur (MAO). En effet, l'usage de boîtes à rythmes ou de séquenceurs permettant d'expérimenter soi-même l'écriture de rythmes en les écoutant mixés avec d'autres instruments sur des machines a permis aux compositeurs de s'émanciper de la création d'un batteur en lui fournissant une partition contenant les parties exactes qu'ils voulaient entendre sur leur musique.

La batterie a un statut à part dans l'univers de l'AMT puisqu'il s'agit d'instruments sans hauteur (du point de vue harmonique), d'événements sonores auxquels une durée est rarement attribuée et de notations spécifiques (symboles des têtes de notes).

Les applications de l'ADT seraient utiles dans tous les domaines musicaux contenant de la batterie dont certains manquent de partitions, notamment les musiques d'improvisation (jazz, pop) [7].

Mais aussi de manière plus générale dans le domaine du MIR. Si les ordinateurs étaient capables d'analyser la partie de la batterie dans la musique enregistrée, cela permettrait une variété de tâches de traitement de la musique liées au rythme. En particulier, la détection et la classification des événements sonores de la batterie par des méthodes informatiques est considérée comme un problème de recherche important et stimulant dans le domaine plus large de la recherche d'informations musicales [10]. L'ADT est un sujet de recherche crucial pour la compréhension des aspects rythmiques de la musique, et a un impact potentiel sur des domaines plus larges tels que l'éducation musicale et la production musicale.

1.4 Les représentations de la musique

Les données audio

Le fichier WAV⁸ est une instance du Resource Interchange File Format (RIFF) défini par IBM et Microsoft. Le format RIFF agit comme une "enveloppe" pour divers formats de codage audio. Bien qu'un fichier WAV

8. <https://en.wikipedia.org/wiki/WAV>

puisse contenir de l'audio compressé, le format audio WAV le plus courant est l'audio non compressé au format LPCM (linear pulse-code modulation). Le LPCM est également le format de codage audio standard des CD audio, qui stockent des données audio LPCM à deux canaux échantillonnées à 44 100 Hz avec 16 bits par échantillon. Comme le LPCM n'est pas compressé et conserve tous les échantillons d'une piste audio, les utilisateurs professionnels ou les experts en audio peuvent utiliser le format WAV avec l'audio LPCM pour obtenir une qualité audio maximale.

Les données MIDI

Le MIDI⁹ (Musical Instrument Digital Interface) est une norme technique qui décrit un protocole de communication, une interface numérique et des connecteurs électriques permettant de connecter une grande variété d'instruments de musique électroniques, d'ordinateurs et d'appareils audio connexes pour jouer, éditer et enregistrer de la musique.

Les données midi sont représentées sous forme de piano-roll. Chaque points sur la figure 1.2 est appelé « évènement MIDI » :



FIGURE 1.2 – Exemple évènements avec durée

Chaque évènement MIDI rassemble un ensemble d'informations sur la hauteur, la durée, le volume, etc. . . :

9. <https://en.wikipedia.org/wiki/MIDI>

Protocol	Event
Property	Value
Type	Note On/Off Event
On Tick	15812
Off Tick	15905
Duration	93
Note	45
Velocity	89
Channel	9

FIGURE 1.3 – Critère pour un évènement

Pour la batterie, les évènements sont considérés sans durée, nous ignorons donc les offsets (« Off Event »), les « Off Tick » et les « Duration ». Le *channel* ne nous sera pas utile non plus.

Ici, définir Tick et channel.

Voici un exemple de piano-roll midi pour la batterie :



FIGURE 1.4 – Exemple évènements sans durée

On observe que toutes les durées sont identiques.

Les partitions



FIGURE 1.5 – Exemple de partition de piano

Une partition de musique¹⁰ est un document qui porte la représentation systématique du langage musical sous forme écrite. Cette représentation est appelée transcription et elle sert à traduire les quatre caractéristiques du son musical :

- la hauteur;
- la durée;
- l'intensité;
- le timbre.

Ainsi que de leurs combinaisons appelées à former l'ossature de l'œuvre musicale dans son déroulement temporel, à la fois :

- diachronique (succession des instants, ce qui constitue en musique la mélodie);
- et synchronique (simultanéité des sons, c'est-à-dire l'harmonie).

1.4.1 Le format MusicXML

MusicXML est un format de fichier basé sur XML pour représenter la notation musicale occidentale. Ce format est ouvert, entièrement documenté et peut être utilisé librement dans le cadre de l'accord de spécification finale de la communauté du W3C.

Un des avantages de ce format est qu'il peut être converti aussi bien en données MIDI qu'en partition musicale, ce qui en fait une interface homme/machine.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
  "-//Recordare//DTD MusicXML 3.1 Partwise//EN"
  "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

FIGURE 1.6 – MusicXML

10. [https://fr.wikipedia.org/wiki/Partition_\(musique\)](https://fr.wikipedia.org/wiki/Partition_(musique))

Le figure 1.6 représente un do en clef de sol de la durée d'une ronde sur une mesure en 4/4.

Conclusion

Dans ce chapitre, nous avons établi que le MIR s'intéresse de plus en plus au TAL, et que, par ce biais, il y a des liens possibles entre le langage musical et les langues naturelles, le plus proche étant probablement le phénomène d'écriture des sons de l'un comme de l'autre.

Nous avons également établi que le MIR est né de l'AMT qui est un problème ancien et très difficile et qu'il serait toujours très utile de le résoudre (autant pour l'AMT que pour l'ADT).

Et enfin, nous avons décrit les représentations de la musique nécessaires à la compréhension du présent mémoire, allant du son jusqu'à l'écriture.

ÉTAT DE L'ART

Sommaire

2.1	Monophonique et polyphonique	19
2.2	Audio vers MIDI	20
2.3	MIDI vers partition	21
2.4	Approche linéaire et approche hiérarchique	21

Introduction

Dans ce chapitre, nous observerons les différentes avancées qui ont déjà eu lieu dans le domaine de la transcription automatique de la musique et de la batterie afin de situer notre démarche.

Nous aborderons le passage crucial du monophonique au polyphonique dans la transcription. Nous ferons un point sur les deux grandes parties de l'AMT de bout en bout : de l'audio vers le MIDI puis des données MIDI vers l'écriture d'une partition. Ensuite, nous discuterons des approches linéaires et des approches hiérarchiques.

2.1 Monophonique et polyphonique

Les premiers travaux ont été faits sur l'identification des instruments monophoniques ¹ [7]. Actuellement, le problème de l'estimation automatique de la hauteur des signaux monophoniques peut être considéré comme résolu, mais dans la plupart des contextes musicaux, les instruments sont polyphoniques. L'estimation des hauteurs multiples (détection multi-pitches ou F0 multiples) est le problème central de la création d'un système de transcription de musique polyphonique. Il s'agit de la détection de notes qui peuvent apparaître simultanément et être produites par

1. Instruments produisant une note à la fois, ou plusieurs notes de même durée (monophonie par accord).

plusieurs instruments différents. Ce défi est donc majeur pour la batterie puisque c'est un instrument qui est lui-même constitué de plusieurs instruments (caisse-claire, grosse-caisse, cymbales, toms, etc...). Le fort degré de chevauchement entre les durées ainsi qu'entre les fréquences complique l'identification des instruments polyphoniques. Cette tâche est étroitement liée à la séparation des sources et concerne aussi la séparation des voix. Les performances des systèmes actuels ne sont pas encore suffisantes pour permettre la création d'un système automatisé capable de transcrire de la musique polyphonique sans restrictions sur le degré de polyphonie ou le type d'instrument. Cette question reste donc encore ouverte.

2.2 Audio vers MIDI

Jusqu'à aujourd'hui, les recherches se sont majoritairement concentrées sur le traitement du signal vers la génération du MIDI [11]. Cette partie englobe plusieurs sous-tâches dont la détection multi-pitches, la détection des onset et des offset, l'estimation du tempo, la quantification du rythme, la classification des genres musicaux, etc...

En ADT [10], plusieurs stratégies de répartition pré/post-processing sont possibles pour la détection multi-pitches. Entamer la détection dès le pré-processing, en supprimant les features non-pertinentes pendant la séparation des sources afin d'obtenir une meilleure détection des instruments de la batterie, est une démarche intuitive : supprimer la structure harmonique pour atténuer l'influence des instruments à hauteurs sur la détection grosse-caisse et caisse-claire en est un exemple. Mais certaines études montrent que des expériences similaires ont donné des résultats non-concluants et que la suppression des instruments à hauteurs peut avoir des effets néfastes sur les performances de l'ADT. En outre, les systèmes d'ADT basés sur des RNN ou des NMF font la séparation des sources pendant l'optimisation, ce qui réduit la nécessité de la faire pendant le pré-processing.

Pour la reconnaissance des instruments, une approche possible [12] est de mettre un modèle probabiliste dans l'étape de la classification des événements afin de classer les différents sons de la batterie. Cette méthode permet de se passer de samples audio isolés en modélisant la progression temporelle des features avec un HMM. Les features sont transformés en représentations statistiques indépendantes. L'approche AdaMa [13] est une autre approche de la même catégorie ; elle commence par une estimation initiale des sons de la batterie qui sont itérativement raffinés pour correspondre à (pour matcher) l'enregistrement visé.

2.3 MIDI vers partition

Le plus souvent, lorsque les articles abordent la transcription automatique de bout en bout (de l'audio à la partition), l'appellation « score » (*partition*) désigne un output au format Music XML, ou simplement MIDI. Par exemple, dans [8], la chaîne de traitement va jusqu'à la génération d'une séquence MIDI quantifiée qui est importée dans MuseScore pour en extraire manuellement un fichier MusicXML contenant plusieurs voix.

Seuls quelques travaux récents s'intéressent de près à la création d'outils permettant la génération de partition. Le problème de la conversion d'une séquence d'événements musicaux symboliques en une partition musicale structurée est traité notamment dans [14]. Ce travail, qui vise à résoudre en une fois la quantification du rythme et la production de partition, s'appuie tout au long du processus sur des grammaires génératives qui fournissent un modèle hiérarchique *a priori* des partitions. Les expériences ont des résultats prometteurs, mais il faut relever qu'elle ont été menées avec un ensemble de données composé d'extraits monophoniques ; il reste donc à traiter le passage au polyphonique en couplant le problème de la séparation des voix avec la quantification du rythme.

L'approche de [14] est fondée sur la conviction que la complexité de la structure musicale dépasse les modèles linéaires.

2.4 Approche linéaire et approche hiérarchique

Plusieurs travaux ont d'abord privilégié l'approche stochastique. Par exemple, Shibata *et al.* [8] ont utilisé le modèle de Markov caché (HMM)² pour la reconnaissance de la métrique. Les auteurs utilisent d'abord deux réseaux de neurones profonds, l'un pour la reconnaissance des pitches et l'autre pour la reconnaissance de la vélocité. Pour la dernière couche, la probabilité est obtenue par une fonction sigmoïde. Ils construisent ensuite plusieurs HMM métriques étendus pour la musique polyphonique correspondant à des métriques possibles, puis ils calculent la probabilité maximale pour chaque modèle afin d'obtenir la métrique la plus probable.

2. https://fr.wikipedia.org/wiki/Modèle_de_Markov_caché
https://en.wikipedia.org/wiki/Hidden_Markov_model

- Modèle de Markov **caché** :
 - **Hidden Markov Model (HMM) (Baum, 1965)**
 - Modélisation d'un processus stochastique « **génératif** » :
 - État du système : non connu
 - Connaissance pour chaque état des **probabilités** comme état initial, de **transition** entre états et de **génération** de symboles
 - **Observations** sur ce qu'a « généré » le système



- Applications : physique, reconnaissance de parole, traitement du langage, bio-informatique, finance, etc.

FIGURE 2.1 – HMM

Source : Cours de Damien Nouvel³

L'évaluation finale des résultats de [8] montre qu'il faut rediriger l'attention vers les valeurs des notes, la séparation des voix et d'autres éléments délicats de la partition musicale qui sont significatifs pour l'exécution de la musique. Or, même si la quantification du rythme se fait le plus souvent par la manipulation de données linéaires allant notamment des *real time units* (secondes) vers les *musical time units* (temps, métrique,...), de nombreux travaux suggèrent d'utiliser une approche hiérarchique puisque le langage musical est lui-même structuré.

En effet, l'usage d'arbres syntaxiques est idéale pour représenter le langage musical. Une méthodologie simple pour la description et l'affichage des structures musicales est présentée dans [15]. Les RT y sont évoqués comme permettant une cohésion complète de la notation musicale traditionnelle avec des notations plus complexes. Jacquemard *et al.* [16] propose aussi une représentation formelle du rythme, inspirée de modèles théoriques antérieurs et dont l'objectif est la réécriture de termes. Ils démontrent aussi l'application des arbres de rythmes pour les équivalences rythmiques dans [17]. La réécriture d'arbres, dans un contexte de composition assistée par ordinateur, par exemple, pourrait permettre de suggérer à un utilisateur diverses notations possibles pour une valeur rythmique, avec des complexités différentes.

La nécessité d'une approche hiérarchique pour la production automatique de partition est évoquée dans [14]. Les modèles de grammaire qui y sont exposés sont différents de modèles markoviens linéaires de précédents travaux.

3. <https://damien.nouvel.net/fr/enseignement>

Example: Summertime

FIGURE 2.2 – arbre_jazz

*Représentation arborescente d'une grille harmonique [9]***Conclusion**

La plupart des travaux déjà existants sur l'ADT ont été énumérés par Wu *et al.* [10] qui, pour mieux comprendre la pratique des systèmes d'ADT, se concentrent sur les méthodes basées sur la factorisation matricielle non négative et celles utilisant des réseaux neuronaux récurrents. La majorité de ces recherches se concentre sur des méthodes de calcul pour la détection d'événements sonores de batterie à partir de signaux acoustiques ou sur la séparation entre les événements sonores de batterie avec ceux des autres instruments dans un orchestre ou un groupe de musique [18], ainsi que sur l'extraction de caractéristiques de bas niveau telles que la classe d'instrument et le moment de l'apparition du son. Très peu d'entre eux ont abordé la tâche de générer des partitions de batterie et, même quand le sujet est abordé, l'output final n'est souvent qu'un fichier MIDI ou MusicXML et non une partition écrite.

Il n'existe pas de formalisation de la notation de la batterie ni de réelle génération de partition finale, dont les enjeux principaux seraient :

- 1) le passage du monophonique au polyphonique, comprenant la distinction entre les sons simultanés et les flas ou autres ornements ;
- 2) les choix d'écritures spécifiques à la batterie concernant la séparation des voix et les continuations.

MÉTHODES

Sommaire

3.1	La notation de la batterie	25
3.2	Modélisation pour la transcription	31
3.3	Qparse	32
3.4	Les systèmes	33

Introduction

Dans ce chapitre, nous expliquerons en détails les méthodes que nous avons employées pour l'ADT.

Pour commencer, nous exposerons une description de la notation de la batterie ainsi qu'une modélisation de celle-ci pour la représentation des données rythmiques en arbres syntaxiques. Nous poursuivrons avec une présentation de qparse¹, un outil de transcription qui est développé par Florent Jacquemard (Inria) au sein du laboratoire Cedric au CNAM. Enfin, nous présenterons les systèmes.

3.1 La notation de la batterie



Une figure de note [19] de musique réunit plusieurs critères² :

— Une tête de note :

Sa position sur la portée indique la hauteur de la note. La tête de note peut aussi indiquer une durée.

1. <https://qparse.gitlabpages.inria.fr/>

2. https://fr.wikipedia.org/wiki/Note_de_musique

- Une hampe :
Indicatrice d'appartenance à une voix en fonction de sa direction et indicatrice d'une durée représentée par sa présence ou non (blanche \neq ronde)
- Un crochet : La durée d'une note est divisée par deux à chaque crochet ajouté à la hampe d'une figure de note.



FIGURE 3.1 – Rapport des figures de notes
[19]

La figure 3.1 montre les rapports de durée entre les figures de notes. Plus les durées sont longues, plus elles sont marquées par la tête de note (la note carrée fait deux fois la durée d'une ronde) ou la présence ou non de la hampe. À partir de la noire (3ème lignes en partant du haut), on ajoute un crochet à la hampe d'une figure de notes pour diviser sa durée par 2. Les notes à crochet (croche, double-croche, triple...) peuvent être reliées ou non par des ligatures (Voir les 4 dernière lignes de la figure 3.1).

Batterie : hauteurs et têtes de notes

Pour la transcriptions, nous proposons une notation inspirée du recueil de pièces pour batterie de J.-F. Juskowiak [20] et des méthodes de batterie Agostini [21], car nous trouvons la position des éléments cohérente et intuitive.

En effet, les hauteurs sur la portée représentent :

- La hauteur physique des instruments :
La caisse claire est centrale sur la portée et sur la batterie (au niveau de la ceinture, elle conditionne l'écart entre les pédales et aussi la position de tous les instruments basiques d'une batterie).
Tous ce qui en dessous de la caisse-claire sur la portée est en dessous de la caisse-claire sur la batterie (pédales, tom basse);
Tous ce qui est au dessus de la caisse-claire sur la portée, l'est

aussi sur la batterie.

- La hauteur des instruments en terme de fréquences :
Sauf pour le charley au pied et si l'on sépare en trois groupes (gc, toms, cymbales), de bas en haut, les instruments vont dans du plus grave au plus aigu.

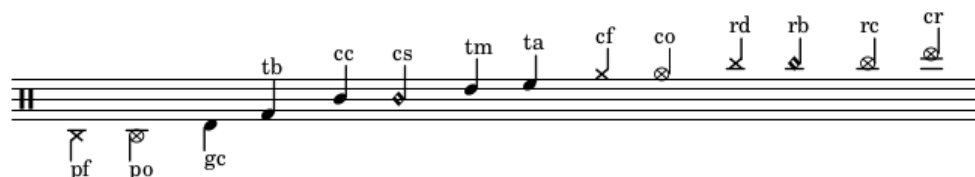


FIGURE 3.2 – Hauteur et têtes de notes

Les noms des instruments correspondant aux codes des notes de la figure 3.2 sont dans le tableau 3.1.

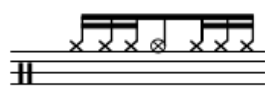
Batterie : durées

Comme nous venons de la voir, la majorité des instruments de la batterie sont représentés par les têtes des notes. Par conséquent, l'usage des symboles rythmiques concernant la tête de note ne pourront pas être utilisées. Ceci est valable aussi pour la présence ou non de la hampe puisque ce phénomène n'existe qu'avec les têtes de notes de type cercle-vide (opposition blanche-ronde). L'usage des blanches existe dans certaines partitions de batterie [ref??system drum ou rythmique bin n.1] mais cela reste des cas très rares. Certains logiciels permettent de faire des blanches avec des symboles spécifiques à la batterie ou aux percussions mais leur lecture reste peu aisée et leur utilisation pour la batterie est rarissime également.

La durée d'une note peut être allongée par divers symboles :

- Le point ;
- La liaison.

Ces symboles ne seront utiles que pour l'écriture des ouvertures de charley. Le charley est le seul instrument de la batterie dont la durée est quantifiée (les cymbales attrapées à la main peuvent l'être aussi mais cela est très rare.)



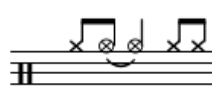
Exemple 1



Exemple 2



Exemple 3



Exemple 4

FIGURE 3.3 – Point et liaison

L'écriture de la batterie doit faire ressortir la pulsation. La première chose à prendre en compte pour analyser la figure 3.3 est donc la nécessité de regrouper les notes par temps à l'aide des ligatures.

exemple 1 : ouverture de charley quantifiée mais pas notes pas regroupées par temps.

exemple 2 : bieeeen !

exemple 3 et exemple 4 : les deux exemples sont valables mais le deuxième est le plus souvent utilisé car plus intuitif (regroupement par temps).

En cas de nécessité de rallonger la durée d'une note pour la batterie, on privilégiera la liaison.

Batterie : silences

	la pause	la demi-pause	le soupir (2)	le demi-soupir	le quart de soupir	le huitième de soupir	le seizième de soupir
Silences							
Notes							
	la ronde	la blanche	la noire	la croche	la double croche	la triple croche	la quadruple croche

FIGURE 3.4 – Les silences

Les silences sont parfois utilisées pour quantifier les ouvertures de charley. Les fermetures du charley sont notées soit par un silence (correspondant à une fermeture de la pédale), soit par un écrasement de l'ouverture

par un autre coup de charley fermé, au pied ou à la main. Physiquement, le charley est fermé par une pression du pied sur la pédale de charley. Dans les fichiers MIDI, cette pression est traduite par un charley joué au pied. Mais dans une vraie partition, cette écriture ne traduirait pas ce que le batteur doit penser.

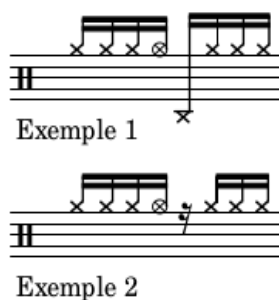


FIGURE 3.5 – Silence joué

L'exemple 1 de la figure 3.5 montre ce qui est écrit dans les données MIDI et l'exemple 2 montre ce que le batteur doit penser en lisant la partition. Il faut aussi prendre en compte l'écriture surchargée que l'exemple 1 donnerait avec une partition comprenant plusieurs voix et plusieurs instruments jouant simultanément.

Batterie : équivalence rythmique

Pour les instruments mélodiques, la liaison et le point sont les deux seules possibilités en cas d'équivalence rythmique pour des notes dont la durée de l'une à l'autre est ininterrompue. Mais pour la batterie, à part pour les ouvertures de charley (voir section 3.1), les durées des notes n'ont pas d'importance. L'usage des silences pour combler la distance rythmique entre deux notes devient donc possible.

Ceci pris en compte, et étant donné que les indications de durée dans les têtes de notes sont peu recommandées (voir section 3.1), l'écriture à l'aide de silences sera privilégiée comme indication de durée sauf dans les cas où cela reste impossible. Ce choix a pour but de n'avoir qu'une manière d'écrire toutes les notes, que leurs têtes de notes soit modifiées ou non.



FIGURE 3.6 – Équivalence

Sur la figure 3.6, il faudra choisir la notation de la deuxième ou de la troisième mesure.

Nuances



FIGURE 3.7 – Nuances

Bien expliquer les accents, remplacer p et f par g et a
 ⇒ nuance VS articulation

Voix

Plusieurs écritures sont possibles pour un même rythme :



FIGURE 3.8 – Séparation des voix

Sur la figure 3.8, il faudra faire un choix entre les exemples 1, 2 et 3 qui sont trois façon d'écrire la même chose. Ce choix se fera en fonction de la lisibilité, de quelles instruments auront des phrasés plus ou moins chargé et/ou variés, auquel cas on les mettra dans une seule voix afin de ne pas charger la partition, etc. Ainsi l'arbre syntaxique de départ sera divisé en autant d'instruments qui le constituent et les voix seront regroupées de manière cohérentes.

3.2 Modélisation pour la transcription

Les pitches

Codes	Instruments	Pitches
cf	charley-main-fermé	22, 42
co	charley-main-ouvert	26
pf	charley-pied-fermé	44
rd	ride	51
rb	ride-cloche (bell)	53
rc	ride-crash	59
cr	crash	55
cc	caisse-claire	38, 40
cs	cross-stick	37
ta	tom-alto	48, 50
tm	tom-medium	45, 47
tb	tom-basse	43, 58
gc	grosse-caisse	36

TABLE 3.1 – Pitches et instruments

Pas de charley pied ouvert. . .

La vélocité

Codes	Instruments	Pitches	Vélocité
cop	charley-main-ouvert	46	?

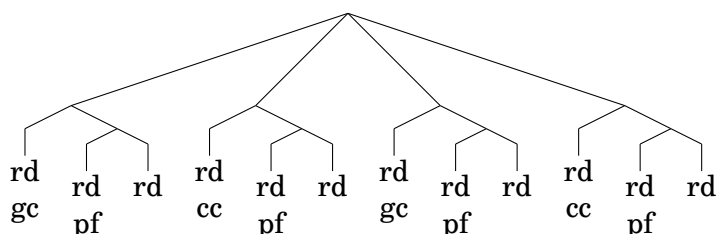
TABLE 3.2 – Vélocité et nuances

Nous ne prendrons en compte la vélocité que pour la cc, les toms et les cymbales jouées aux mains. Les nuances de grosse caisse et charley aux pieds sont le plus souvent insignifiantes, elles ne sont marquées sur le figure qu'à titre indicatif. Si la vélocité est en dessous de 40, il s'agit de ghost-notes : la tête de note devra être entouré de parenthèses et le suffixe *p* (*piano*) devra être ajouté au codes de l'instrument. (Voir ccp ci-dessus.) Si la vélocité est au dessus de 90, il s'agit de notes accentuées : le symbole « > » et le suffixe *f* (*forte*) devra être ajouté au codes de l'instrument. (Voir ccf ci-dessus.) Lorsque la vélocité va de 40 à 89, on considèrera le volume comme normal et aucun symbole supplémentaire ne sera ajouté à la note.

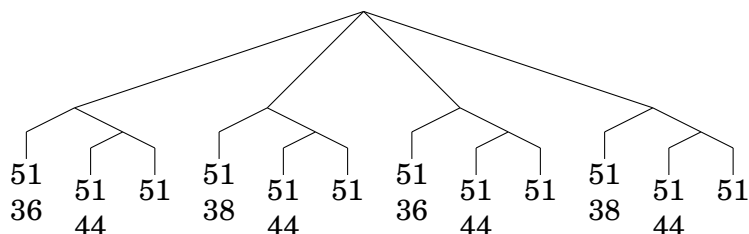
Le charley de pitch 46 est considéré comme le charley ouvert joué à la main sur le haut de la cymbale mais souvent, ça correspond au geste « tranche-olive » de la baguette lorsque le batteur accentue avec la tranche et joue moins fort avec l'olive sur le plat de la cymbale. Je vais dans un premier temps considérer le pitch comme **charley-main-ouvert-piano** (ghost-note)

Les arbres de rythmes

Voici une représentation de la figure 3.8 en arbre de rythme avec les codes de chaque instrument :



Ci-dessous, le même arbre dont les codes des instruments sont remplacés par leurs données midi respectives :



Cet arbre représente un rythme unique dont les possibilités de notation sur une partition sont théoriquement multiples. Les trois exemples de la figure 3.4 peuvent-être représentés par les arbres ci-dessus.

3.3 Qparse

Florent Jacquemard(Inria), Philippe Rigaux (CNAM), Masahiko Sakai (Nagoya U.), Francesco Foscari (PhD, CNAM), Clement Poncelet (Salzburg U.) *Mettre ici un schéma de la chaîne de traitement de qparse (workflow)*

Pb du MIDI avec qparse entrée ON-OFF \Rightarrow sortie 1 seul symbole.

Qparse produit une partition musicale en prenant en entrée une performance musicale symbolique (par exemple un fichier MIDI) et un automate

à arbre pondéré décrivant un langage de rythmes préférés (grammaire pondérée). La quantification des rythmes est basée sur des algorithmes d'analyse syntaxique applicables sur des automates arborescents.³ En entrée : midi (séquence d'événements datés (piano roll) accompagné d'une grammaire pondérée)

⇒ parsing

⇒ global parsing tree

⇒ RI (Représentation Intermédiaire) arbres locaux par instruments

⇒ Sortie (xml, mei, lilypond, . . .)

Minimiser la distance entre le midi et la représentation en arbre.

Un des problèmes de Qparse était qu'il était limité au monophonique. ⇒ *Expliquer ici les limites du monophonique. . .*

La grammaire pondérée

La grammaire pondérée qui accompagne le MIDI en input est une grammaire hors-contexte pondérée. Chaque règle comporte un poid qui sert à favoriser certains rythmes plutôt que d'autres.⁴

Le parsing

Le parsing du midi donné en input crée une représentation symbolique sous forme d'arbre de rythme.

Ici ⇒ exemple avec :

3bars_fill_groove-016.mid ⇒ arbre

La réécriture

Séparation des voix

Simplification

Ici, description basique des règles de réécriture

3.4 Les systèmes

Un système est la combinaison d'un ou plusieurs éléments qui jouent un rythme en boucle (motif) et d'un autre élément qui joue un texte rythmique variable mais respectant les règles propre au système (gamme).

Système = motif + gamme/texte

motif = rythmes coordonnés joués avec 2 ou 3 membres en boucle (reparti

3. <https://qparse.gitlabpages.inria.fr>

4. <https://qparse.gitlabpages.inria.fr/docs/scientific/>

sur 1 ou 2 voix)

gamme/texte = rythme irrégulier joué avec un seul membre sur le motif (Réparti sur 1 voix). La gamme d'un système considère l'ensemble des combinaisons que le batteur pourrait rencontrer en interprétant un texte rythmique à l'aide du système.

Nous partirons de propositions génériques de systèmes (environs trois systèmes dans différents styles de batterie) que nous tenterons de détecter dans le jeu de données groove.

Quatre systèmes standards :

- binaire
- ternaire (shuffle, afro, rock)
- jazz
- afro-cubain

Nous travaillerons aussi sur la détection de répétitions sur plusieurs mesures afin de pouvoir corriger des erreurs sur une des mesures qui aurait dû être identique aux autres mais qui présente des différences.

Intérêt des systèmes

Détection d'indication de mesure et choix de grammaire pondérée

Il faut prendre en compte l'existence potentielle de plusieurs grammaires (*un fichier wta par grammaire*) chacune dédiée à un type de contenu MIDI. Le choix d'une grammaire pondérée doit être fait avant le parsing puisque qparse prends en entrée un fichier MIDI et un fichier wta.

Pour les expériences effectuées avec le Groove MIDI Data Set, le style et l'indication de mesure sont récupérables par les noms de fichiers MIDI, mais il faudra par la suite les trouver automatiquement sans autres indications que les données MIDI elles-mêmes.

En conséquence, les motifs des systèmes devront être recherchés sur l'input (*fichiers MIDI*) avant le lancement du parsing, afin de déterminer la métrique en amont en vu de sélectionner la grammaire pondérée (*fichier wta*) adéquate pour le parsing. Nous pensons que cette tâche devrait être effectuée en Machine Learning. **Les systèmes devront être matchés sur l'input MIDI**

- Définir une métrique ;
- Réécriture : séparation des voix ;
- Réécriture : Set de règles spécifiques de simplification.

Il faudra créer un ensemble de systèmes comprenant leurs règles spécifiques de réécriture (séparation des voix et simplifications). 3 grandes catégories :

Systèmes	Métriques	Subdivisions	Possibles	nb voix
binaires	simple	doubles-croches	triolet, sextolet	2
jazz	simple	triolet	croches et doubles-croches	2
ternaires	complexe	croches	duolet, quartelet	2
afros-cubains	simple	croches	-	3

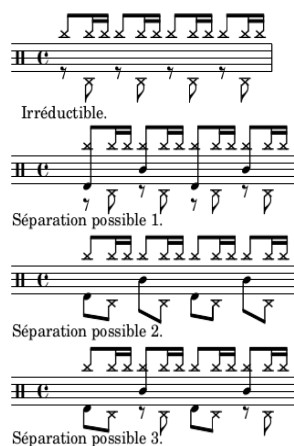
TABLE 3.3 – Systèmes

- ternaire (mesures complexes, principalement croches, noire pointée, duolets et quartelets possibles)
- afro-cubain (mesure)
- Tout transcrire avec LilyPond et en arbres d'analyse syntaxique.
- Créer les arbres de voix séparées.
- Écrire les règles de réécriture.
- Créer les arbres de voix séparées simplifiés (rewriting).

Pour la **séparation des voix** et la **définition des métriques**, nous nous intéresserons principalement à la partie *motif* des systèmes qui seront présentés. La partie *texte* nous intéressera plus pour les **combinaisons de réécritures**.

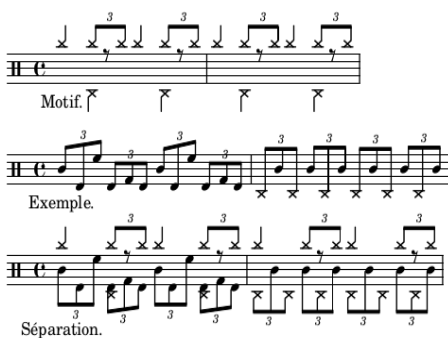
Réécriture — Pour la séparation des voix

Motif 4-4 binaire



Ici, le système est construit sur un modèle rock en 4/4 : after-beat sur les 2 et 4 avec un choix de répartition des cymbales type fast-jazz. Le système est constitué par défaut du motif ride/ch-pf/cc et d'un texte joué à la grosse-caisse. La troisième séparation proposée est privilégiée car elle répartit selon 2 voix, une voix pour les mains (ride + cc) et une voix pour les pieds (ch-pf + gc). Ce choix paraît plus équilibré car deux instruments sont utilisés par voix et plus logique pour le lecteur puisque les mains sont en haut et les pieds en bas.

Motif 4-4 jazz



Dans la plupart des méthodes, le charley n'est pas écrit car considéré comme évident en jazz traditionnel. Ce qui facilite grandement l'écriture : la ride et les crash sur la voix du haut et le reste sur la voix du bas. Ici, le partie prit et de tout écrire. Dans l'exemple ci-dessus, les mesures 1 et 2 combinées avec le *motif* de la première ligne, sont des

cas typiques de la batterie jazz. Tout mettre sur la voix haute serait surchargé. De plus, la grosse caisse entre très souvent dans le flot des combinaisons de toms et de caisse claire et son écriture séparée serait inutilement compliquée et peu intuitive pour le lecteur. Le choix de séparation sera donc de laisser les cymbales en haut et toms, caisse-claire, grosse-caisse et pédale de charley en bas.

Système 4-4 afro-cubain



Pour la reconnaissance de la métrique

12/8 vs 4/4 ternaire

Motif 12/8

Pour les règles de réécriture

Les textes qui accompagnent les motifs étayent toutes les combinaisons d'un systèmes.

Exemples à écrire en arbre :

- SI (pas pf) ET (note sur un temps suivie de note en l'air) :
⇒ (Temps1 : Note pertinente) + (Temps2 : Silence pertinent + Note pertinente.)
- Si (po ou co) déborde sur le temps suivant :
⇒ Liaison car marchera dans tous les cas même la où le point ne marchera pas (voir A2).
- Une blanche sera écrite noir + soupir.

Conclusion

Bilan sur les différentes méthodes employées et la contribution que cela représente.

EXPÉRIMENTATIONS

Sommaire

4.1	Corpus	39
4.2	Analyse MIDI-Audio	41
4.3	Expérimentation théorique d'un système	46
4.4	Développement	50

Introduction

Dans ce chapitre, nous présenterons le corpus, les analyses audio-MIDI. Nous ferons la démonstration d'un modèle théorique de système (implémentable) qui devra être utilisé comme base de connaissances pour obtenir un système plus rapide et une meilleure qualité en sortie. Enfin, nous présenterons les différentes contributions de développement.

4.1 Corpus

Différence de qualité de jeu
groove MIDI dataset

<https://magenta.tensorflow.org/datasets/groove>



Des batteurs pro ont été engagés pour jouer sur un roland td-11

The Groove MIDI Dataset (GMD), has several attributes that distinguish it from existing ones:

- The dataset contains about 13.6 hours, 1,150 MIDI files, and over 22,000 measures of drumming.
- Each performance was played along with a metronome set at a specific tempo by the drummer.
- The data includes performances by a total of 10 drummers, with more than 80% of duration coming from hired professionals. The professionals were able to improvise in a wide range of styles, resulting in a diverse dataset.
- The drummers were instructed to play a mix of long sequences (several minutes of continuous playing) and short beats and fills.
- Each performance is annotated with a genre (provided by the drummer), tempo, and anonymized drummer ID.
- Most of the performances are in 4/4 time, with a few examples from other time signatures.
- Four drummers were asked to record the same set of 10 beats in their own style. These are included in the test set split, labeled `eval-session/groove1-10`.
- In addition to the MIDI recordings that are the primary source of data for the experiments in this work, we captured the synthesized audio outputs of the drum set and aligned them to within 2ms of the corresponding MIDI files.

Les métadatas :

The metadata file (`info.csv`) has the following fields for every MIDI/WAV pair:

Field	Description
drummer	An anonymous string ID for the drummer of the performance.
session	A string ID for the recording session (unique per drummer).
id	A unique string ID for the performance.
style	A string style for the performance formatted as "<primary>/<secondary>". The primary style comes from the Genre List below.
bpm	An integer tempo in beats per minute for the performance.
beat_type	Either "beat" or "fill"
time_signature	The time signature for the performance formatted as "<numerator>-<denominator>".
midi_filename	Relative path to the MIDI file.
audio_filename	Relative path to the WAV file (if present).
duration	The float duration in seconds (of the MIDI).
split	The predefined split the performance is a part of. One of "train", "validation", or "test".

Genre List: afrobeat, afrocuban, blues, country, dance, funk, gospel, highlife, hiphop, jazz, latin, middleeastern, neworleans, pop, punk, reggae, rock, soul

A train/validation/test split configuration is provided for easier comparison of model accuracy on various tasks.

Split	Beats	Fills	Measures (approx.)	Hits	Duration (minutes)
Train	378	519	17752	357618	648.5
Validation	48	76	2269	44044	82.2
Test	77	52	2193	43832	84.3
Total	503	647	22214	445494	815.0

Détails (entre autres tensorflow avec le dataset) à : <https://magenta.tensorflow.org/datasets/groove#license>
écouter le dataset groove

4.2 Analyse MIDI-Audio

Ces analyses ont été faites essentiellement dans le cadre de transcription manuelles à partir de fichiers MIDI et Audio du Groove MIDI Data Set. Les partitions manuelles sont éditées à l'aide de lilypond¹. Les transcriptions automatiques sont générées avec MuseScore².

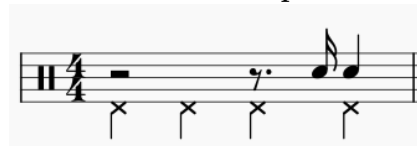
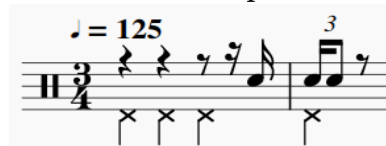
-
1. <http://lilypond.org/>
 2. <https://musescore.com/>

Comparaisons de transcriptions

Transcription manuelle VS transcription automatique :

drummer_01/session3 — 10_rock-folk_90_beat_4-4

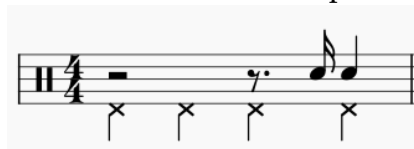
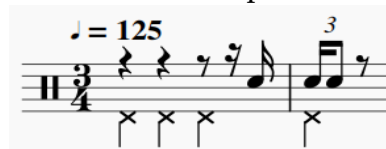
Fichier midi vers partition avec musescore ⇒ Transcription manuelle



drum-

mer_01/session3 — 10_rock-folk_90_beat_4-4

Fichier midi vers partition avec musescore ⇒ Transcription manuelle



- Erreur d'indication de mesure ;
- Mauvaise transcription d'une noire.

La noire du 4ème temps se retrouve sur le premier temps de la mesure suivante et elle se transforme en un triolet de double croches dont seules les deux premières seraient jouées.

drummer_01/session3 — 10_rock-folk_90_beat_4-4

Fichier midi vers partition avec musescore ⇒ Transcription manuelle



- Erreur de quantification : les doubles croches ont été interprétées en quintolet ;

drummer_01/session3 — 2_jazz-swing_185_beat_4-4

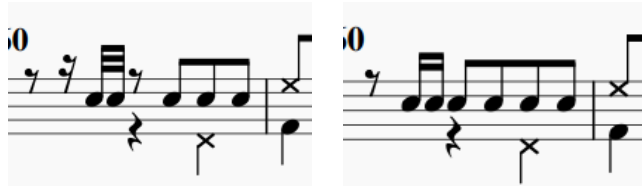
Fichier midi vers partition avec musescore ⇒ Transcription manuelle



- L'indication de mesure est correcte mais tout a été décalé d'un temps car la première noire sur la caisse claire est jouée sur le 4ème temps et non sur le premier temps de la deuxième mesure comme l'indique la transcription de musescore.
- Les toms basses des 1er et 2ème temps de la mesure musescore auraient dû être sur les temps et non décalés d'une double croche vers la droite.

drummer_01/session1 — 1_funk_80_beat_4-4

Fichier midi vers partition avec musescore ⇒ Transcription manuelle



- On dirait que lorsque certaines notes sont proches, elles se resserrent et suppriment celles qui aurait dû être sur le temps.
- Erreur d'indication de mesure ;
- Mauvaise transcription d'une noire.

La noire du 4ème temps se retrouve sur le premier temps de la mesure suivante et elle se transforme en un triolet de double croches dont seules les deux premières seraient jouées.

drummer_01/session3 — 10_rock-folk_90_beat_4-4

Fichier midi vers partition avec musescore ⇒ Transcription manuelle



- Erreur de quantification : les doubles croches ont été interprétées en quintolet ;

drummer_01/session3 — 2_jazz-swing_185_beat_4-4

Fichier midi vers partition avec musescore ⇒ Transcription manuelle



- L'indication de mesure est correcte mais tout a été décalé d'un temps car la première noire sur la caisse claire est jouée sur le 4ème temps et non sur le premier temps de la deuxième mesure comme l'indique la transcription de musescore.
- Les toms basses des 1er et 2ème temps de la mesure musescore auraient dû être sur les temps et non décalés d'une double croche vers la droite.

drummer_01/session1 — 1_funk_80_beat_4-4

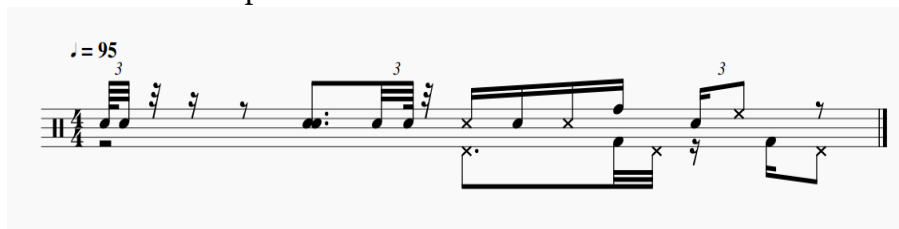
Fichier midi vers partition avec musescore ⇒ Transcription manuelle



- On dirait que lorsque certaines notes sont proches, elles se resserrent et suppriment celles qui aurait dû être sur le temps.

Exemple avec des flas

Fichier midi vers partition avec musescore :



Transcription manuelle :



MuseScore donne un aperçu de l'état de l'art pour la transcription de la

batterie.

Transcription de partition

FIGURE 4.1 – référence 1

Il s'agit d'une partition d'un 4/4 binaire dont le fichier MIDI est annoncé dans le groove-dataset de style «jazz-funk» probablement en raison de la ride de type shabada rapide (le ternaire devient binaire avec la vitesse) combiné avec l'after-beat de type rock (caisse-claire sur les deux et quatre).

La transcription de cette partition a occasionné plusieurs remarque :
- vélocité, place des accents, etc. . .

4.3 Expérimentation théorique d'un système

Motifs

Les motifs 1 et 2 peuvent être extraits de la figure 3.8. Ces deux motifs sont très classiques et seront réutilisables aussi dans d'autres contextes.

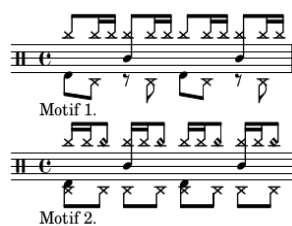


FIGURE 4.2 – Motifs

Le motif 1 est joué jusqu'à la mesure 18 avec des variations et des breaks.
Le motif 2 est joué des mesures 23 à 28.

Gammes

Débit croches.



Débit doubles-croches.



Systèmes

Motif 1, texte 1a.

Motif 1, texte 1b.

Motif 1, texte 2a.

Motif 1, texte 2b.

Motif 1, texte 2c.

Motif 2, texte 1a.

Motif 2, texte 1b.

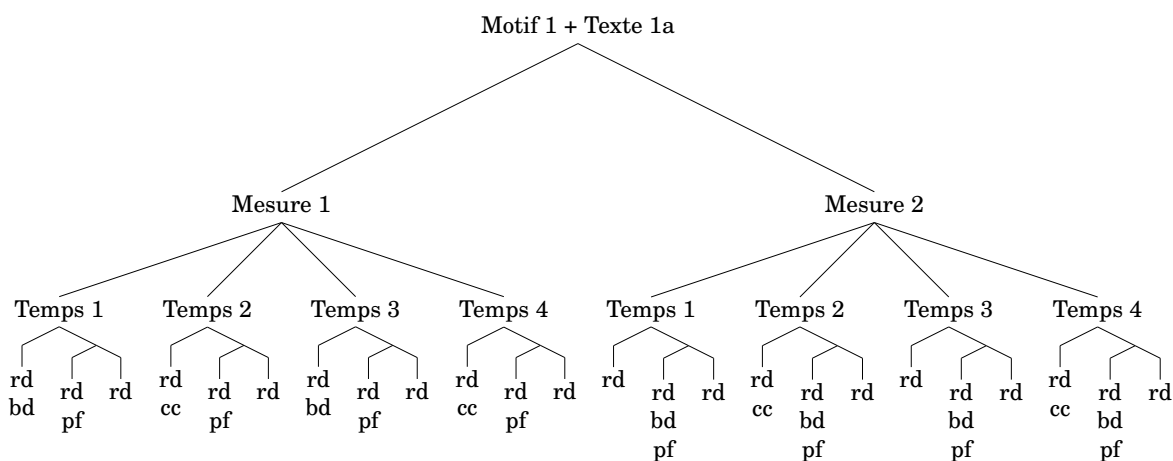
Motif 2, texte 2a.

Motif 2, texte 2b.

Motif 2, texte 2c.

Démonstration

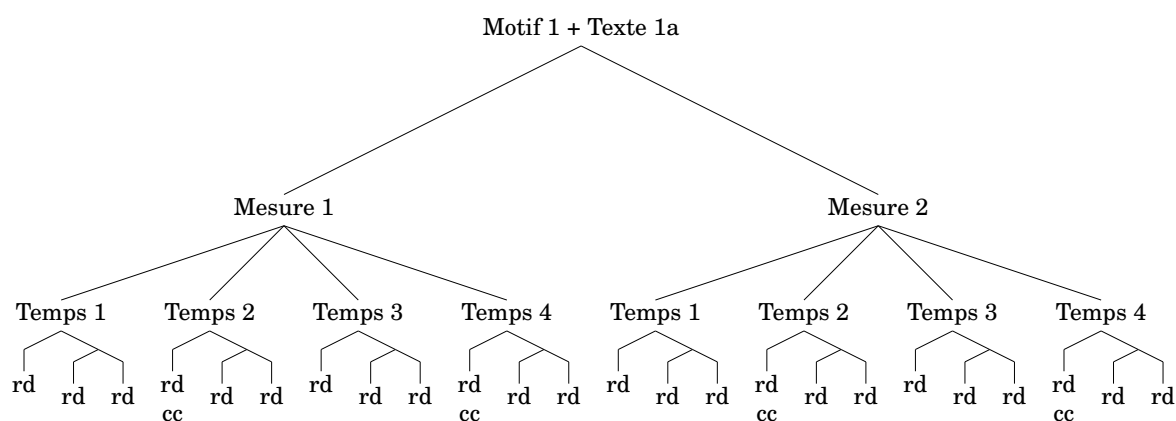
Représentation des systèmes en arbres de rythmes



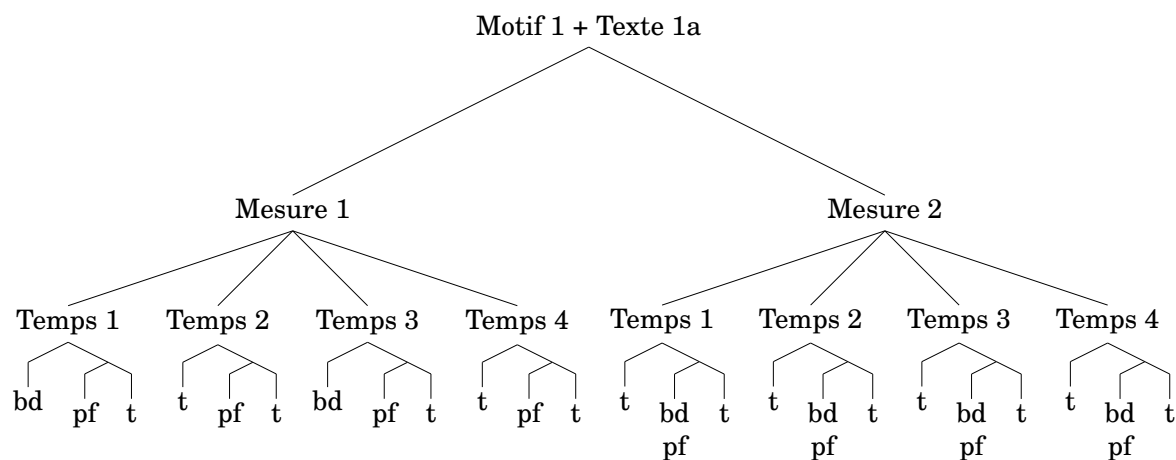
Réécriture Règles établies par le système

Séparation des voix

Voix haute

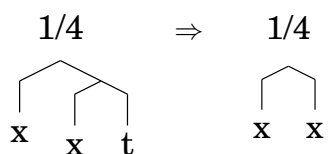
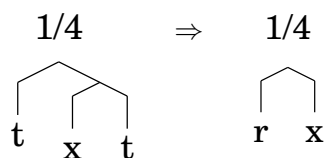
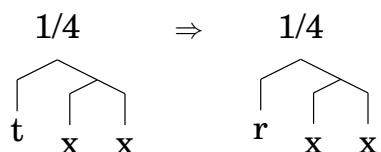
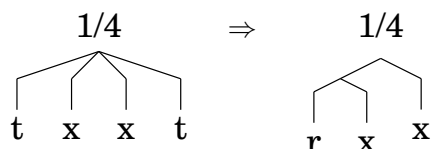
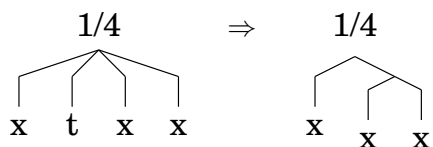
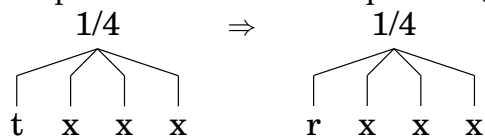


Voix basse



Règles de simplifications pour le 4/4 binaire

Simplifier l'écriture de chaque voix (*Règles établis par le système*)



4.4 Développement

DrumCode

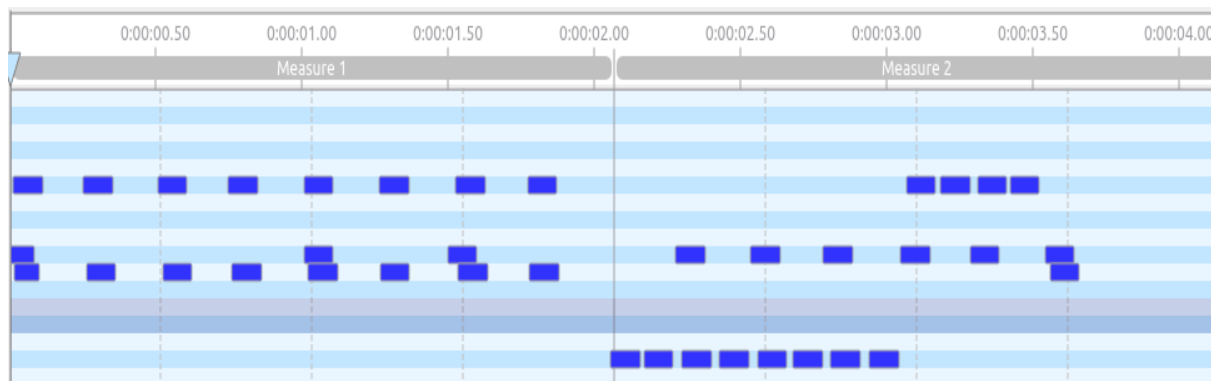
Adaptation de la modélisation pour la transcription en cpp.

Tests unitaires sur les Jams

Les Jams permettent de passer du monophonique au polyphonique.

Parsing

Tests effectués avec le fichier midi suivant :



Un premier test convaincant est effectué avec la grammaire suivante :

```
// bar level
0 -> C0 1
0 -> E1 1
0 -> U4(1, 1, 1, 1) 1
```

```
// half bar level
9 -> C0 1
9 -> E1 1
```

```
// beat level
1 -> C0 1
1 -> E1 1
1 -> T2(2, 2) 1
1 -> T4(4, 4, 4, 4) 1
```

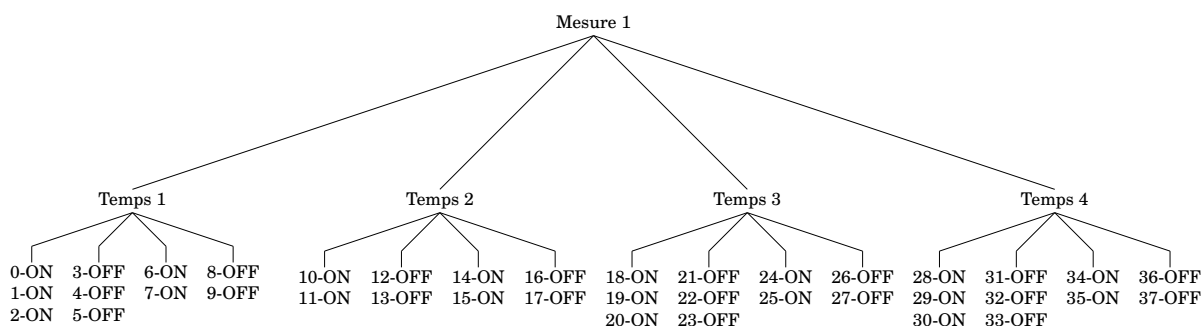
```
// croche level
2 -> C0 1
2 -> E1 1
```

```
// double level
4 -> C0 1
4 -> E1 1
```

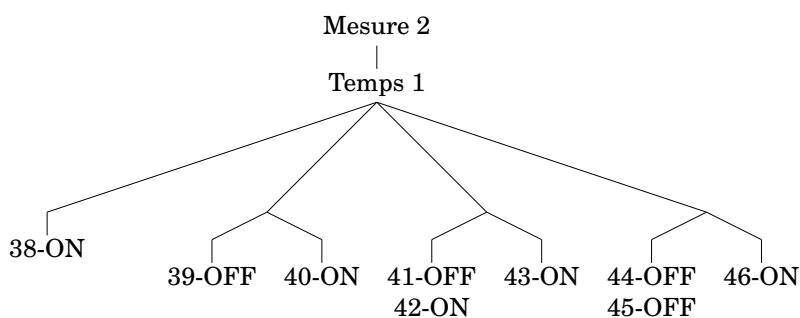
```
4 -> E2 1
4 -> T2(6, 6) 1
```

```
// triple level
6 -> E1 1
```

Cette grammaire sépare les ligatures par temps au niveau de la mesure. Puis, au niveau du temps, elle autorise les divisions par deux (croches) et par quatre (doubles-croches). Tous les poids sont réglés sur 1. L'arbre de parsing en résultant est considéré comme « convaincant » car il découpe correctement les mesures et les temps.



Les temps de la première mesure du fichier MIDI sont bien quantifié mais ceux de la deuxième mesure présentent quelques défauts de quantification visibles dès le premier temps.



Les Onsets sont correctement triés au niveau des doubles croches mais certaines doubles croches sont inutilement subdivisées en triples croches (les 2ème, 3ème et 4ème doubles croches sur le premier temps ci-dessus).

2ème exemple :

Après une augmentation du poids des triples croches dans la grammaire

(monté de 1 à 5) et une baisse de tous les autres poids (descendu de 1 à 0.5), et mis à part le troisième temps de la 2ème mesure, tous les Onsets sont bien triés et aucuns ne sont subdivisés.

Conclusion

Conclusion de ce chapitre.

DISCUSSION

Sommaire

5.1	Travaux réalisés	55
5.1.1	Développer la notation	55
5.1.2	La modélisation	55
5.1.3	Le jeu de système	55
5.2	Travaux non-réalisés	55
5.3	Travaux futures	56

Introduction

Dans ce chapitre, nous discuterons sur la pertinence de l'ensemble des choix qui ont été faits. Nous ferons un bilan des différentes avancées qui ont été faites ou non et nous tenterons d'en expliquer la ou les raisons.

5.1 Travaux réalisés

Faire une auto-critique des travaux réalisés.

5.1.1 Développer la notation

5.1.2 La modélisation

5.1.3 Le jeu de système

5.2 Travaux non-réalisés

Expliquer pourquoi ces travaux n'ont pas pu être réalisés.

- implémenter un pattern. . .
⇒ manque de temps ?

- La partie résultat est manquante car :
 - ⇒ Sujet très difficile ;
 - ⇒ Matcher les motifs peut être fait ultérieurement ;
Mais ce travail aurait été indispensable pour obtenir une quantité de résultats qui justifieraient une évaluation automatique permettant de faire des graphiques.
- L'évaluation fut entièrement manuelle car :
 - ⇒ Très dure automatiquement : il faut comparer 2 partitions (réf VS output)

5.3 Travaux futures

- Le ternaire jazz (voir expérience 2)
- Reconnaissance d'un motif sur le MIDI
Reconnaître un motif (système) sur une mesure de l'input (un fichier midi représentant des données audios)
 - ⇒ Motif (système) reconnu : true ou false
 - Si true :
 - Choisir la grammaire correspondante ;
 - Parser le MIDI ;
 - Appliquer les règles de réécritures (Séparation des voix et simplification)

Conclusion

Sujet passionnant mais difficile. Obtenir la totalité des critères pour le mémoire n'aurait pas pu être fait sans bâcler. Une base solide spécifique à la batterie a été générée. Elle sera un bon point de départ pour les travaux futurs dont plusieurs propositions sont énoncés dans le présent document.

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons traité de la problématique de la transcription automatique de la batterie. Son objectif était de transcrire, à partir de leur représentation symbolique MIDI, des performances de batteur de différents niveaux et dans différents styles en partitions écrites.

Nous avons avancé sur le parsing des données MIDI établissant un processus de regroupement des événements MIDI qui nous a permis de faire la transition du monophonique vers le polyphonique. Une des données importante de ce processus était de différencier les nature des notes d'un *accord*, notamment de distinguer lorsque 2 notes constituent un *accord* ou un *fla*.

Nous avons établis des *grammaires pondérées* pour le parsing qui correspondent respectivement à des métriques spécifiques. Celles-ci étant sélectionnables en amont du parsing, soit par indication des noms des fichiers MIDI, soit par reconnaissance de la métrique avec une approche dictionnaire de patterns prédéfinis ¹ qu'il serait pertinent de mettre en œuvre en machine learning.

Nous avons démontré que l'usage des *systèmes* élimine un grand nombre de calcul lors de la réécriture. Pour la séparation des voix grâce au motif d'un système et pour la simplification grâce aux gammes du motif d'un système. Nous avons aussi montré comment, dans des travaux futurs, un système dont le motif serait reconnu en amont dans un fichier MIDI pourrait prédéfinir le choix d'une grammaire par la reconnaissance d'une métrique et ainsi améliorer le parsing et accélérer les choix ultérieurs dans la chaîne de traitement en terme de réécriture.

Il sera également intéressant d'étudier comment l'utilisation de LM peut améliorer les résultats de l'AM, voir [2], et ouvrir la voie à la génération entièrement automatisée de partitions de batterie et au problème général de l'AMT de bout en bout.[7]

1. *Motifs* dans les *systèmes* de la présente proposition.

BIBLIOGRAPHIE

- [1] H. C. Longuet-Higgins. Perception of melodies. 1976. – Cité pages 9 et 12.
- [2] Meinard Müller. *Fundamentals of Music Processing*. 01 2015. – Cité page 9.
- [3] Caroline Traube. Quelle place pour la science au sein de la musicologie aujourd’hui? *Circuit*, 24(2) :41–49, 2014. – Cité page 10.
- [4] Bénédicte Poulin-Charronnat and Pierre Perruchet. Les interactions entre les traitements de la musique et du langage. *La Lettre des Neurosciences*, 58 :24–26, 2018. – Cité page 11.
- [5] Mikaela Keller, Kamil Akesbi, Lorenzo Moreira, and Louis Bigo. Techniques de traitement automatique du langage naturel appliquées aux représentations symboliques musicales. In *JIM 2021 - Journées d’Informatique Musicale*, Virtual, France, July 2021. – Cité page 11.
- [6] Junyan Jiang, Gus Xia, and Taylor Berg-Kirkpatrick. Discovering music relations with sequential attention. In *NLP4MUSA*, 2020. – Cité page 11.
- [7] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription : Challenges and future directions. *Journal of Intelligent Information Systems*, 41, 12 2013. – Cité pages 12, 14, 19 et 57.
- [8] Kentaro Shibata, Eita Nakamura, and Kazuyoshi Yoshii. Non-local musical statistics as guides for audio-to-score piano transcription. *Information Sciences*, 566 :262–280, 2021. – Cité pages 12, 21 et 22.
- [9] Daniel Harasim, Christoph Finkensiep, Petter Ericson, Timothy J O’Donnell, and Martin Rohrmeier. The jazz harmony treebank. – Cité pages 12 et 23.
- [10] Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinard Müller, and Alexander Lerch. A review of automatic drum transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9) :1457–1483, 2018. – Cité pages 14, 20 et 23.

- [11] Moshakwa Malatji. Automatic music transcription for two instruments based variable q-transform and deep learning methods, 10 2020. – Cité page 20.
- [12] Antti J. Eronen. Musical instrument recognition using ica-based transform of features and discriminatively trained hmms. *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, 2 :133–136 vol.2, 2003. – Cité page 20.
- [13] Hiroshi G. Okuno Kazuyoshi Yoshii, Masataka Goto. Automatic drum sound description for real-world music using template adaptation and matching methods. *International Conference on Music Information Retrieval (ISMIR)*, pages 184–191, 2004. – Cité page 20.
- [14] Francesco Foscarin, Florent Jacquemard, Philippe Rigaux, and Masahiko Sakai. A Parse-based Framework for Coupled Rhythm Quantization and Score Structuring. In *MCM 2019 - Mathematics and Computation in Music*, volume Lecture Notes in Computer Science of *Proceedings of the Seventh International Conference on Mathematics and Computation in Music (MCM 2019)*, Madrid, Spain, June 2019. Springer. – Cité pages 21 et 22.
- [15] C. Agon, K. Haddad, and G. Assayag. Representation and rendering of rhythm structures. In *Proceedings of the First International Symposium on Cyber Worlds (CW'02)*, CW '02, page 109, USA, 2002. IEEE Computer Society. – Cité page 22.
- [16] Florent Jacquemard, Pierre Donat-Bouillud, and Jean Bresson. A Term Rewriting Based Structural Theory of Rhythm Notation. Research report, ANR-13-JS02-0004-01 - EFFICACe, March 2015. – Cité page 22.
- [17] Florent Jacquemard, Adrien Ycart, and Masahiko Sakai. Generating equivalent rhythmic notations based on rhythm tree languages. In *Third International Conference on Technologies for Music Notation and Representation (TENOR)*, Coruña, Spain, May 2017. Helena Lopez Palma and Mike Solomon. – Cité page 22.
- [18] R. Marxer and J. Janer. Study of regularizations and constraints in nmf-based drums monaural separation. In *International Conference on Digital Audio Effects Conference (DAFx-13)*, Maynooth, Ireland, 02/09/2013 2013. – Cité page 23.
- [19] A. Danhauser. *Théorie de la musique*. Edition Henry Lemoine, 41 rue Bayen - 75017 Paris, Édition revue et augmentée - 1996 edition, 1996. – Cité pages 25 et 26.
- [20] J.-F. Juskowiak. *Rythmiques binaires 2*. Alphonse Leduc, Editions Musicales, 175, rue Saint-Honoré, 75040 Paris, 1989. – Cité page 26.

-
- [21] Dante Agostini. *Méthode de batterie, Vol. 3*. Dante Agostini, 21, rue Jean Anouilh, 77330 Ozoir-la-Ferrière, 1977. – Cité page 26.

