ROSENHEIMER INFORMATIKPREIS
INF-MASTER

GUITAR TABLATURE GENERATION FROM MIDI SIGNALS USING TRANSFORMERS

ANNA HAMBERGER (MASTERSTUDIUM INFORMATIK)

Betreuer: Prof. Dr. Jochen Schmidt, Prof. Dr. Marcel Tilly

Diese Masterarbeit stellt einen neuartigen Ansatz zur automatischen Transkription von Gitarrentabulaturen aus MIDI-Daten vor. Im Gegensatz zu bisherigen Methoden, die meist auf Suchalgorithmen basieren, kommt erstmals ein T5-Transformermodell zum Einsatz. Aufgrund der polyphonen Struktur der Gitarre und der vielfältigen Griffmöglichkeiten stellt die Tabulaturerzeugung eine besondere Herausforderung dar. Die Arbeit nutzt neben dem bestehenden DadaGP-Datensatz auch zwei neue Datensätze – GuitarToday und Leduc – und untersucht verschiedene Modellierungsstrategien. Die entwickelten Modelle, darunter eine Variante mit Kapodaster-Konditionierung, erzielen vielversprechende Ergebnisse und übertreffen bestehende Ansätze. Damit leistet die Arbeit einen wichtigen Beitrag zur automatisierten Musikverarbeitung und zur Anwendung von Transformermodellen im Bereich der Gitarrentranskription.

Motivation

Music is a universal form of expression with both auditory and written representations. Transcribing music from audio to written notation – known as Automatic Music Transcription (AMT) – is a complex, time-consuming task, particularly for polyphonic music like that played on the guitar. While AMT research has made progress in transcribing music into standard notation, guitarists often rely on tablature, which includes essential information about string and fret positions not captured in traditional sheet music. However, converting music to tablature presents unique challenges due to the multiple possible fingerings for the same pitch and the need for physically playable fingerings. This thesis proposes a novel transformer-based approach to automatically generate guitar tablature from symbolic MIDI input, aiming to bridge the gap between standard notation and instrument-specific transcription.

Basics of Natural Language Processing

Natural Language Processing (NLP) enables computers to interpret and generate human language and plays a crucial role in modern AI applications such as translation systems and conversational agents [1]. This thesis applies NLP techniques to a novel domain by framing the generation of guitar tablature from MIDI sequences as a text-to-text translation task, treating both as distinct musical 'languages'. The work builds on the Transformer architecture, a neural network model that uses self-attention mechanisms and parallel processing to model long-range dependencies in sequences [2]. In particular, the thesis employs the T5 (Text-to-Text Transfer Transformer) model [3], which unifies NLP tasks into a common text-to-text format.

Basics of Music Processing

Music processing involves the transformation of musical information between different representations. In Western music notation, pitch is described using pitch class, octave, and accidentals, and is visualized on a five-line staff [4]. The MIDI (Musical Instrument Digital Interface) format encodes musical events such as pitch, duration, and dynamics in digital form. Unlike traditional notation, MIDI represents notes with numerical values (0–127) and uses event-based messages to indicate note-on and note-off events [5]. Guitar tablature offers a practical notation system tailored for guitarists [6]. It displays string and fret positions directly, allowing even players without formal training to perform music. Figure 1 shows a classical music notation and the corresponding guitar tablature. The same pitch can be played in several ways on the guitar due to its string-fret redundancy, making transcription more complex.

Transcribing guitar tablature from other formats, such as sheet music or MIDI, presents challenges. Multiple valid fingerings exist for the same note or chord, and choosing an optimal one depends on musical quality, anatomical feasibility, and stylistic preferences. Additional factors such as tuning variations, use of a capo, and instrument-specific constraints further influence the resulting tablature. A successful transcription must therefore balance musical accuracy with physical playability.

Guitar Tablature Generation with Transformers

The availability of high-quality datasets plays a crucial role in the development and evaluation of machine learning models. Three symbolic GuitarPro [7] datasets are used. GuitarToday [8] includes 363 beginner-oriented pieces with standard tuning, simple rhythms, and a focus on open strings and low frets. DadaGP [9] offers over 2,300 guitar tracks from a larger, genre-spanning collection, featuring more complex rhythms, a wider pitch and fret range, and mostly 24-fret guitars. Leduc [10] provides 232 jazz tablatures with mid-range pitch emphasis, broader fret usage, and characteristic jazz complexity. Together, these datasets support training models capable of producing accurate, stylistically varied, and playable tablature.

Figure 1: Beginning of a tablature for 'Smoke on the Water' by Deep Purple. Source: GuitarToday Dataset, rendered in MuseScore

The data preprocessing converts GuitarPro files into a structured text format suitable for transformer models. First, datasets are cleaned and acoustic guitar tracks were isolated using MIDI instrument IDs and guitar-related keywords. Metadata is completed and standardized, duplicates are removed, and the dataset is split into training, validation, and test sets while preserving key musical features. Next, GuitarPro files are converted to MIDI. Only relevant guitar tracks are processed, and each note's key attributes – start time, end time, pitch, string, and fret – are extracted. Finally, the data is encoded into tokens. Eleven different encodings are developed to explore various levels of abstraction. Each encoding is tokenized and converted into numerical sequences for training the transformer models.

The T5 model is used for tablature generation due to its strong performance in sequence-to-sequence tasks, treating MIDI-to-tab transcription as a translation problem. The pre-trained t5-small version is fine-tuned on the prepared datasets. Initial tests show that the default hyperparameters work well.

To enable fair comparison across different encodings, two post-processing methods are applied to model outputs. The token-based method checks token sequences for structure, inserting error markers for invalid tokens, while the value-based method extracts pitch, string, and fret information, discarding incomplete or invalid entries. No corrections are made – outputs reflect the model's raw predictions.

To assess model performance beyond standard NLP metrics, several music-specific measures are introduced, focusing on musical accuracy and guitar playability. Pitch Preservation verifies that the predicted notes match the original MIDI pitches, allowing for alternate but correct fret-string positions and String-Fret Mapping evaluates how accurately the model reproduces the ground truth's string and fret positions. These metrics help compare model outputs in terms of both correctness and real-world usability for guitarists.

Experiments and Results

A series of experiments was conducted to evaluate different strategies. Each experiment tested specific hypotheses about model performance, and the resulting insights significantly shaped the final optimized model. The baseline performance provided a low-level reference point for randomly generated tablatures, which produce the correct pitch but do not produce meaningful playable tablatures. Key findings included the importance of proper MIDI-to-text encoding and the effectiveness of a 512-token input length during training. Inference performance was optimized using overlapping sequences of 20 notes. Conditional setups (use of capo, tuning) were tested but limited by data imbalance, which was partly mitigated through capo-based data augmentation. A smaller custom model outperformed the pre-trained T5-small.

Two optimized model versions were created, one with standard settings and one with capo conditions. Both outperformed the baseline, showing competitive or superior results to existing methods.

Bibliography

[1] R. S. T. Lee, Natural Language Processing: A Textbook with Python Implementation. Singapore: Springer Nature Singapore, 2023.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Neural Information Processing Systems, pp. 5998 – 6008, 2017.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.

[4] C. Hamm, "Open Music Theory: Introduction to Western Musical Notation." https://viva.pressbooks.pub/openmusictheory/chapter/introduction-to-western-musical-notation, 2021. Accessed: 2024-11-14.

[5] T. M. M. Association, "Standard midi files 1.0." https://midi.org/general-midi-level-1, 1996. Accessed: 2024-11-14.

[6] R. Denyer, The Guitar Handbook. London: Pan Books, 1992.

[7] "Guitar Pro - Tab Editor Software for Guitar, Bass, Drum, Piano and more." https://www.guitar-pro.com/. Accessed: 2024-11-14.

[8] Harry, "GuitarToday | creating music lesson | Patreon." https://www.patreon.com/guitartoday/about. Accessed: 2024-11-14.

[9] P. Sarmento, A. Kumar, C. J. Carr, Z. Zukowski, M. Barthet, and Y.-H. Yang, "DadaGP: A Dataset of Tokenized GuitarPro Songs for Sequence Models," in Proceedings of the 22th International Society for Music Information Retrieval Conference, pp. 610–617, 2021.

[10] "François Leduc Online Library." https://www.francoisleduconlinelibrary.com/. Accessed: 2024-11-14.