Research Proposal on Rhythm Pattern as Word Embedding

Researchers:

Lu Tongyu Yan Lucheng

lutongyucn@foxmail.com dun_1989@163.com

Motivations

- How to elegantly integrate musical prior knowledge with symbolic algorithm composing models?

Two main considerations:

1. To design model structures
2. To design musical element representations

A typical example of consideration 1:

Cope, D. (1989). *Experiments in musical intelligence (EMI): Non-linear linguistic-based composition. Interface, 18(1-2), 117–139.* doi:10.1080/09298218908570541

Motivations

- How to elegantly integrate musical prior knowledge with symbolic algorithm composing models?

Two main considerations:

1. To design model structures
2. **To design musical element representations**

How to design such representations?

Think music as a kind of language!

Motivations

- How to elegantly integrate musical prior knowledge with symbolic algorithm composing models?

Two main considerations:

1. To design model structures
2. To design musical element representations

Music is emotional & logical!

An instructive introduction for music starters on how to perceive music as language (video in Chinese)

Reference: Wiwi Kuan TED https://www.youtube.com/watch?v=hkMLzn6Gjv4

Motivations

- Music as Language vs. Natural Language

Common: music has syntactic structures which are similar to natural language

Music	Natural Language
Note or chord	Character
Measure	Word
Phrase	Sentence
Période	Paragraph
Movement	Passage

Motivations

- Music as Language vs. Natural Language

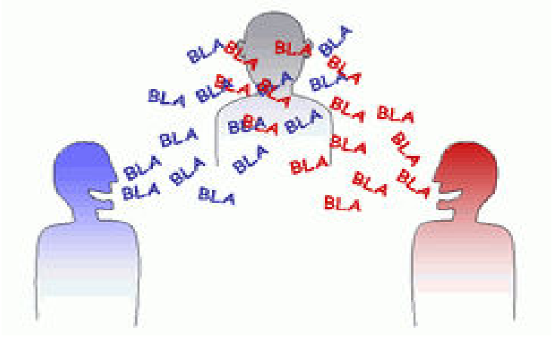
However, there are several differences between them:

Difference #1:
Music is an art of harmony!

Normally, music is polyphonic (multi-dimension of parts).

Natural languages cannot be polyphonic!

http://clipart-library.com/clipart/6Tp5XRgjc.htm



https://instinctink.wordpress.com/2016/03/14/cocktail-party-effect/

The Cocktail Party Phenomenon: Humans can sift out audio information which is inconsequent

*Bronkhorst, Adelbert W. (2000). "The Cocktail Party Phenomenon: A Review on Speech Intelligibility in Multiple-Talker Conditions" (PDF). Acta Acustica United with Acustica. **86**: 117–128. Retrieved 2010-04-18.*

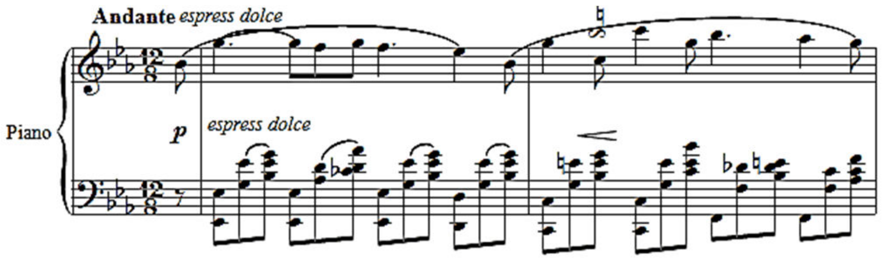
Motivations

- Music as Language vs. Natural Language

However, there are several differences between them:

Difference #2:
Music does not convey concrete semantics!

It is hard to translate music into identical natural language.



"Chopin was feeling good"

"Chopin was feeling sad"

…

"Musical semantics is a paradoxical matter."…
"Unlike any natural language, music resists translation."

JP Swain. (1996). "The range of musical semantics" (PDF). The Journal of aesthetics and art criticism, 1996 - JSTOR

Motivations

- Music as Language vs. Natural Language

By the way, I think the semantic concept in this paper is actually syntactic

Leave difference #1 to chord syntactics.

e.g. chord embeddings

e.g. harmonization

Phil Chen and Edward Xu. *(2016).* "CS224N Project Report: From Note2Vec to Chord2Vec" *(PDF).* pdfs.semanticscholar.org

S Madjiheurem, L Qu, C Walder. (2016). "Chord2vec: Learning musical chord embeddings" (PDF). Proceedings of the constructive

CH.Chuan, K.Agres, D.Herremans *. (2020). "From context to concept: exploring semantic relationships in music with word2vec" (PDF).* Neural Computing and Applications, 2020 - Springer

Leave difference #2 to musical conditional generation with description and emotion analysis. (text2music and music2text)

As far as I know, the problem of conditional generation of music with text description is not explored. However, text2image has been achieved.

S Reed, Z Akata, X Yan, L Logeswaran. (2016). "Generative adversarial text to image synthesis" (PDF). International Society for Music Information Retrieval 2011

Motivations

- How to elegantly integrate musical prior knowledge with symbolic algorithm composing models?

Two main considerations:

1. To design model structures
2. **To design musical element representations**

- We start from doing **rhythm syntactics**.

(As far as I know, there is no such study on rhythm syntactics with the concepts of NLP, although there are existing projects on rhythm generation)

For example:
Aaron Levisohn and Philippe Pasquier. *(2008).* "BeatBender: subsumption architecture for autonomous rhythm generation" *(PDF)*. Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology December 2008 Pages 51-58 https://doi.org/10.1145/1501750.1501762

- In other words, we take rhythm syntactics as part of musical prior knowledge.

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence
Période	Paragraph
Movement	Passage

Packing

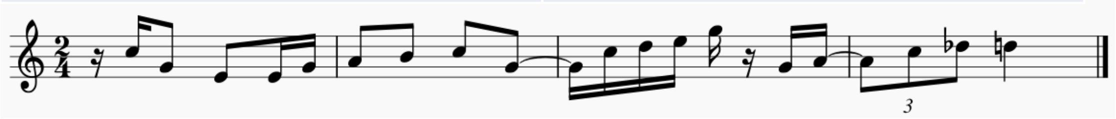
In a larger scheme, we may not care about a single note, but the rhythm pattern as a whole.

Analogy made by Yan: notes are like molecules in a cell, while a rhythm pattern is like a cell as a whole. When we study the function of a organ, we seldom study the molecules.

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



←Just for illustration

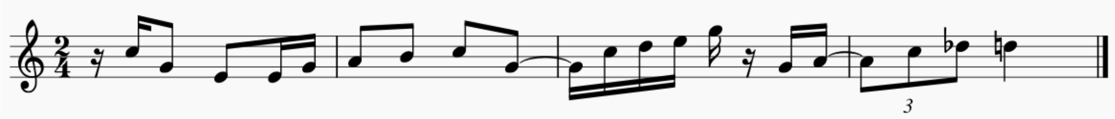
Similar to ↔ ["<BOS>","I", "am", "feeling", "well", "<EOS>"]

How to represent the rhythm in the given melody as words?

Rhythm embedding: word2vec

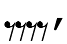
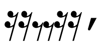
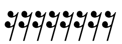
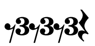
- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



How to represent the rhythm in the given melody as words?

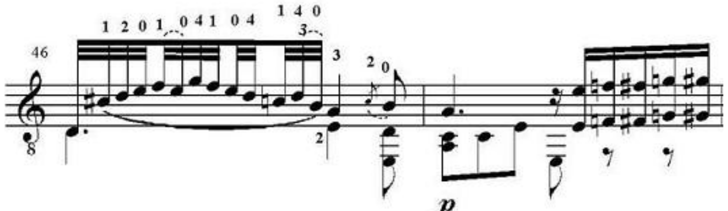
A naïve way (proposed by Yan): exhaust all normal mode of rhythm patterns and build a dictionary.

Rhythm_dict = {0: “<BOS>”, 1: “<EOS>”, 2: “2/4”, 3: , 4: , 5: , 6: }

Rhythm = [0,2,4,3,5,6,1]

However, we cannot exhaust every possible rhythm pattern.
It is also hard to consider rhythms in different meters.

e.g. can we consider **this** rhythm pattern in advance? →

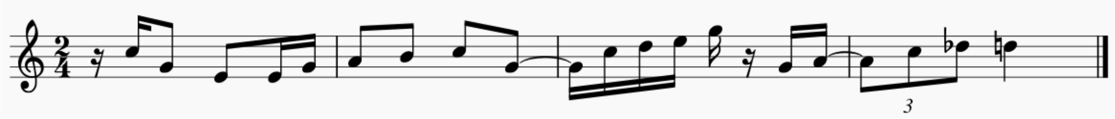


Paganini Romance Piu tosto Largo Amorosamente

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



How to represent the rhythm in the given melody as words?

Another way (inspired by Yan, proposed by Lu): treat notes as characters.

♪ = Note with ½ beats = "N0.500"

𝄿 = Rest with ¼ beats = "R0.250"

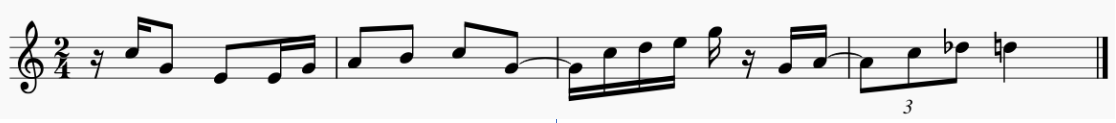
…

This method aims to let our model learn to automatically summarize different rhythm patterns.

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



How to represent the rhythm in the given melody as words?

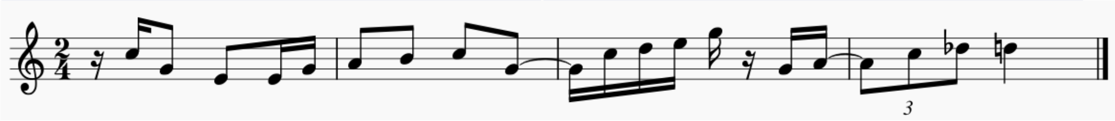
```
Rhythm = ["<BOS>", "|2/4",
"R0.250,N0.250,N0.500,N0.250,N0.250|2/4",
"N0.500,N0.500,N0.500,N0.500|2/4",
"H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4",
"H0.333,N0.333,N0.333,N1.000|2/4", "<EOS>"]
```

'Hold'

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



How to represent the rhythm in the given melody as words?

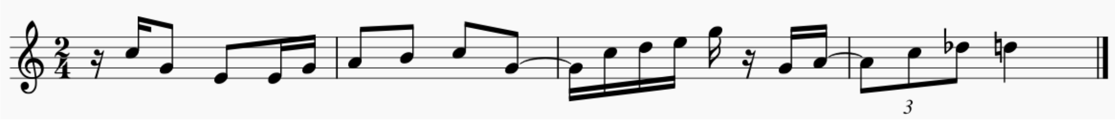
```
Rhythm = [0,2,4,3,5,6,1]

Rhythm_vocabulary = {0:“<BOS>”, 1:“<EOS>”, 2:“|2/4”,
3:“N0.500,N0.500,N0.500,N0.500|2/4”,
4:“R0.250,N0.250,N0.500,N0.250,N0.250|2/4”,
5:“H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4”,
6:“H0.333,N0.333,N0.333,N1.000|2/4”}
```

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



How to represent the rhythm in the given melody as words?

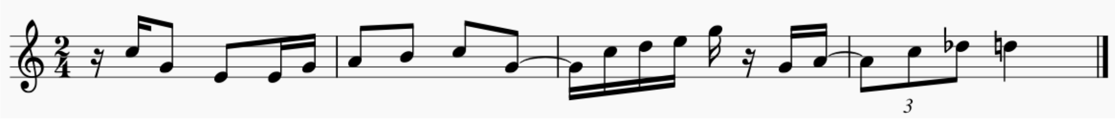
We can enlarge our rhythm vocabulary by feeding more pieces to our model!

```
Rhythm_vocabulary = {0:"<BOS>", 1:"<EOS>", 2:"|2/4",
3:"N0.500,N0.500,N0.500,N0.500|2/4",
4:"R0.250,N0.250,N0.500,N0.250,N0.250|2/4",
5:"H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4",
6:"H0.333,N0.333,N0.333,N1.000|2/4"}
```

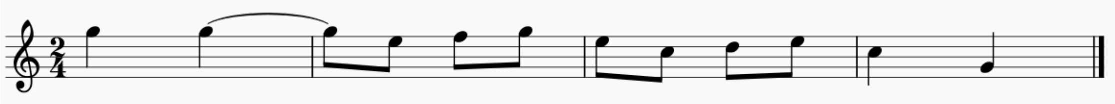
Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



How to represent the rhythm in the given melody as words?



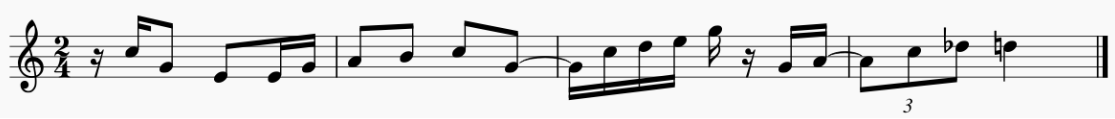
← Faure: Berceuse

```
Rhythm_vocabulary = {0:"<BOS>", 1:"<EOS>", 2:"|2/4",
3:"N0.500,N0.500,N0.500,N0.500|2/4",
4:"R0.250,N0.250,N0.500,N0.250,N0.250|2/4",
5:"H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4",
6:"H0.333,N0.333,N0.333,N1.000|2/4"}
```

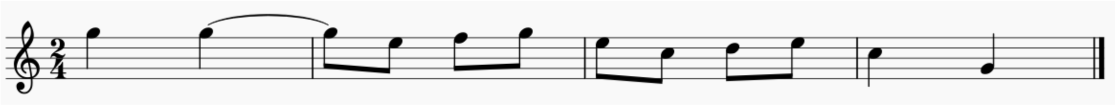
Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word



How to represent the rhythm in the given melody as words?



← Faure: Berceuse

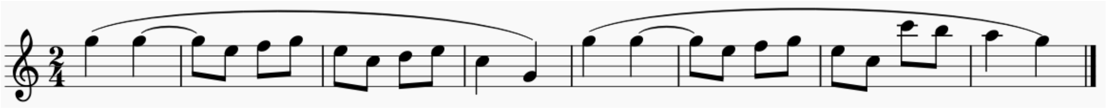
```
Rhythm_vocabulary = {0:“<BOS>”, 1:“<EOS>”, 2:“|2/4”,
3:“N0.500,N0.500,N0.500,N0.500|2/4”,
4:“R0.250,N0.250,N0.500,N0.250,N0.250|2/4”,
5:“H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4”,
6:“H0.333,N0.333,N0.333,N1.000|2/4”,
7:“H0.500,N0.500,N0.500,N0.500|2/4”, 8:“N1.000, N1.000|2/4”}
```

Rhythm embedding: word2vec

- Treat phrases as periods

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence

How to represent phrases?



← Faure: Berceuse

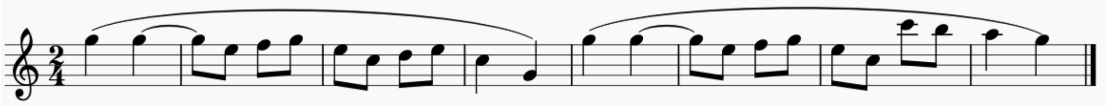
```
Rhythm_vocabulary = {0:“<BOS>”, 1:“<EOS>”, 2:“|2/4”,
3:“N0.500,N0.500,N0.500,N0.500|2/4”,
4:“R0.250,N0.250,N0.500,N0.250,N0.250|2/4”,
5:“H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4”,
6:“H0.333,N0.333,N0.333,N1.000|2/4”,
7:“H0.500,N0.500,N0.500,N0.500|2/4”, 8:“N1.000, N1.000|2/4”}
```

Rhythm embedding: word2vec

- Treat phrases as periods

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence

How to represent phrases?



← Faure: Berceuse

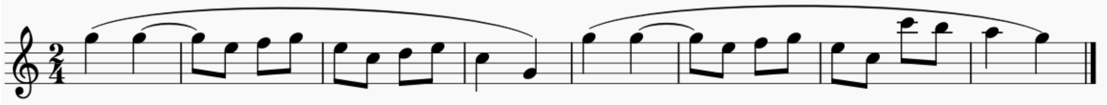
```
Rhythm_vocabulary = {0:“<BOS>”, 1:“<EOS>”, 2:“|2/4”,
3:“N0.500,N0.500,N0.500,N0.500|2/4”,
4:“R0.250,N0.250,N0.500,N0.250,N0.250|2/4”,
5:“H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4”,
6:“H0.333,N0.333,N0.333,N1.000|2/4”,
7:“H0.500,N0.500,N0.500,N0.500|2/4”, 8:“N1.000, N1.000|2/4”,
9:“<BREATH>”}
```

Rhythm embedding: word2vec

- Treat phrases as periods

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence

How to represent phrases?



← Faure: Berceuse

```
Berceuse_rhythm = [0,2,8,7,3,8,9,8,7,3,8,1]
```

```
Rhythm_vocabulary = {0:“<BOS>”, 1:“<EOS>”, 2:“|2/4”,
3:“N0.500,N0.500,N0.500,N0.500|2/4”,
4:“R0.250,N0.250,N0.500,N0.250,N0.250|2/4”,
5:“H0.250,N0.250,N0.250,N0.250,N0.250,R0.250,N0.250, N0.250|2/4”,
6:“H0.333,N0.333,N0.333,N1.000|2/4”,
7:“H0.500,N0.500,N0.500,N0.500|2/4”, 8:“N1.000, N1.000|2/4”,
9:“<BREATH>”}
```

Rhythm embedding: word2vec

- Treat phrases as periods

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence

How to represent phrases?

How to let machine learn to label phrases?

We need labelled data with phrases!

In our following baseline experiment, we did not use <BREATH> to label phrases due to the heavy workload. **←TODO**

Rhythm embedding: word2vec

- Treat phrases as periods

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence

How to represent phrases?

How to let machine learn to label phrases?

Actually, this is a subtask of symbolic MIR. This task is called **music pattern discovery**.

Iris Yuping Ren, Anja Volk, Wouter Swierstra, Remco C. Veltkamp. *(2020).* *"A Computational Evaluation of Musical Pattern Discovery Algorithms" . In Review*

Applications of music pattern discovery include **variation detection**, **theme extraction** and **music segment detection**.

Anja Volk, W. Bas de Haas, Peter van Kranenburg. *(2012).* *"Towards Modelling Variation in Music as Foundation for Similarity" (PDF). Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Science of Music July 23-28, 2012*

Rhythm embedding: word2vec

- Treat phrases as periods

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence

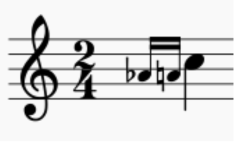
How to represent phrases?

Remaining problems:

1. What if a period ends within a meter?
2. What if we encounter rhythm-rubato?
3. How to deal with grace notes?

Treat <BREATH> as a note and plug it into the string representing a meter.

Use a marker "|RBT" to note that there is a rubato, and count duration for every beat as a word.



← Use markers like 'G+2,N1.000' to mark grace notes like this.

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence
Période	Paragraph
Movement	Passage

Packing

An open question:
What if we still care about notes in the scheme of phrases?

When we pack up the notes/characters into meters/words, we throw away the information conveyed by single elements (i.e. information of single notes/characters).
What if such single-element-information is still important? Are there other methods of representations of rhythm patterns which explicitly retain information of notes? ←**TODO**

Rhythm embedding: word2vec

- Treat rhythm as words

Rhythm	Natural Language
Duration of note	Character
Measure	Word
Phrase	Sentence
Période	Paragraph
Movement	Passage

Packing

My naïve solution: carefully design word embeddings, which take into account of edit distance. Anyway, I am about to talk about rhythm2vector.

An open question:
What if we still care about notes in the scheme of phrases?

For example, there may be *rhythm pattern A* and *rhythm pattern B* which only differ in one note. However, *A* often appears in our database, while *B* only appears once in our database. In common sense, *A* and *B* function in the same way, but our model did not detect this, unless it can consider the information in note-level.

Rhythm embedding: word2vec

- How to represent syntactic meanings of rhythm patterns?

Word embedding (word2vec): a popular method in NLP

article_dataset.txt

Methods e.g. BM25, TF-IDF, trie tree, HMM etc.

Word Division
e.g. nltk, jieba for python

```
[["I", "am", "feeling", "well"],
["Tom", "said", "he", "did", "it"]]
```

Data Preprocess and Build Dictionary

```
[one_hot(["<BOS>","I", "am", "feeling", "well", "<EOS>"]),
one_hot(["<BOS>","Tom", "said", "he", "did", "it", "<EOS>"])]
```

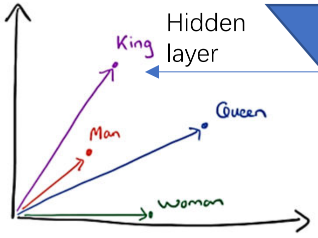
Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

prediction

Hidden layer

Word vectors



rhythm2vec?

waltzes1.mid
waltzes2.mid
waltzes3.mid
waltzes4.mid

① MIDI decoding
e.g. music21 for python

② Main Part Extraction (for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

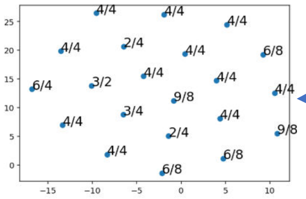
④ Rhythm string encoder and dictionary builder

Data Preprocess

Neural Network

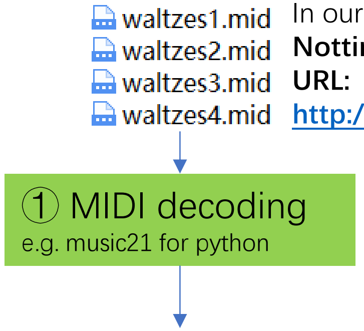
e.g. Prediction based Methods such as CBOW, skip-gram.

prediction



Rhythm embedding: word2vec

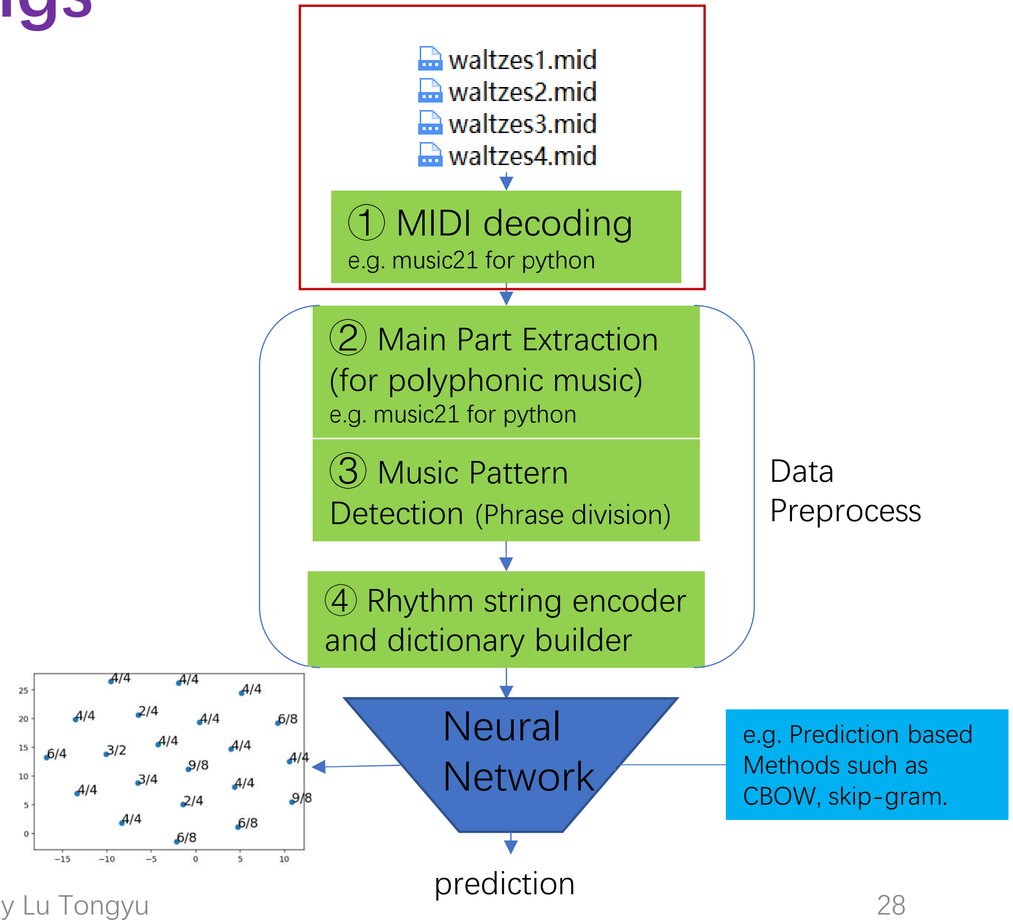
- Details of rhythm embeddings



In our experiment, we use **Nottingham Music Database. URL:** http://abc.sourceforge.net/NMD/

Stream of midi objects.

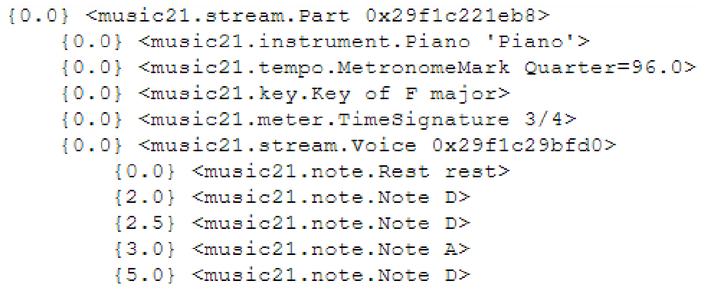
```
{0.0} <music21.stream.Part 0x29f1c221eb8>
    {0.0} <music21.instrument.Piano 'Piano'>
    {0.0} <music21.tempo.MetronomeMark Quarter=96.0>
    {0.0} <music21.key.Key of F major>
    {0.0} <music21.meter.TimeSignature 3/4>
    {0.0} <music21.stream.Voice 0x29f1c29bfd0>
        {0.0} <music21.note.Rest rest>
        {2.0} <music21.note.Note D>
        {2.5} <music21.note.Note D>
        {3.0} <music21.note.Note A>
        {5.0} <music21.note.Note D>
```



Rhythm embedding: word2vec

- Details of rhythm embeddings

Stream of midi objects.

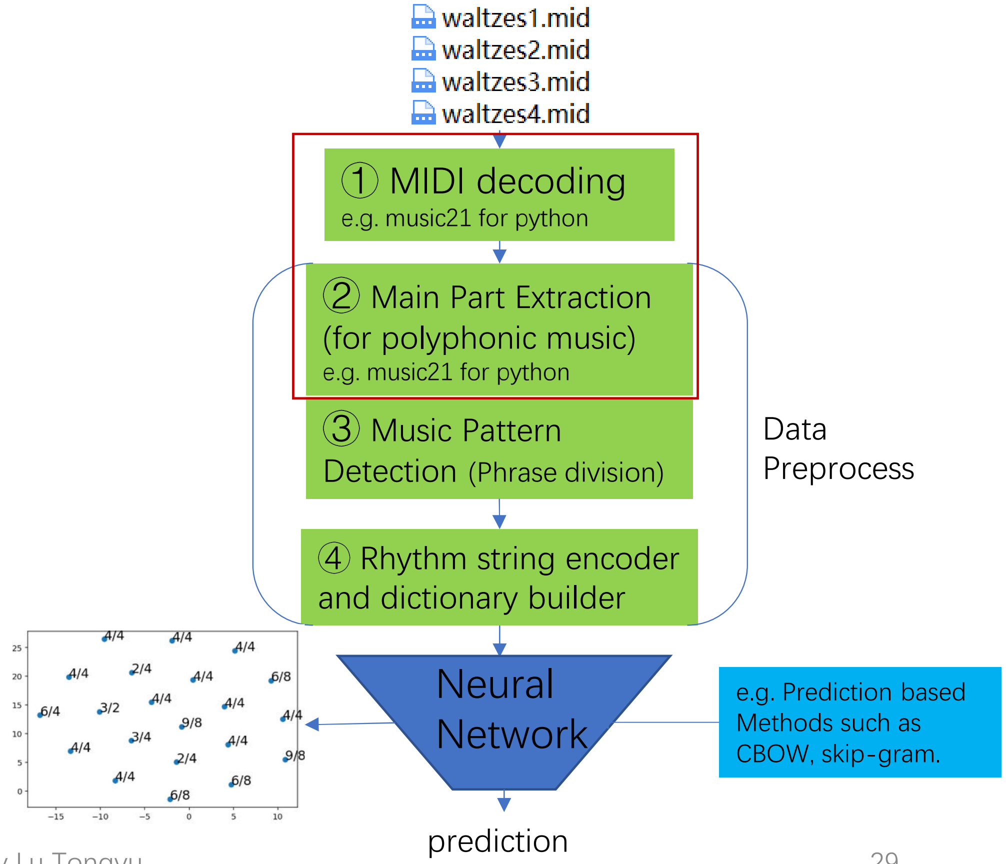


```
{0.0} <music21.stream.Part 0x29f1c221eb8>
    {0.0} <music21.instrument.Piano 'Piano'>
    {0.0} <music21.tempo.MetronomeMark Quarter=96.0>
    {0.0} <music21.key.Key of F major>
    {0.0} <music21.meter.TimeSignature 3/4>
    {0.0} <music21.stream.Voice 0x29f1c29bfd0>
        {0.0} <music21.note.Rest rest>
        {2.0} <music21.note.Note D>
        {2.5} <music21.note.Note D>
        {3.0} <music21.note.Note A>
        {5.0} <music21.note.Note D>
```

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python

This task should be combined with music pattern detection if our music is polyphonic.

When to use main part extraction?
My proposal is as follows:



waltzes1.mid
waltzes2.mid
waltzes3.mid
waltzes4.mid

① MIDI decoding
e.g. music21 for python

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

prediction

Rhythm embedding: word2vec

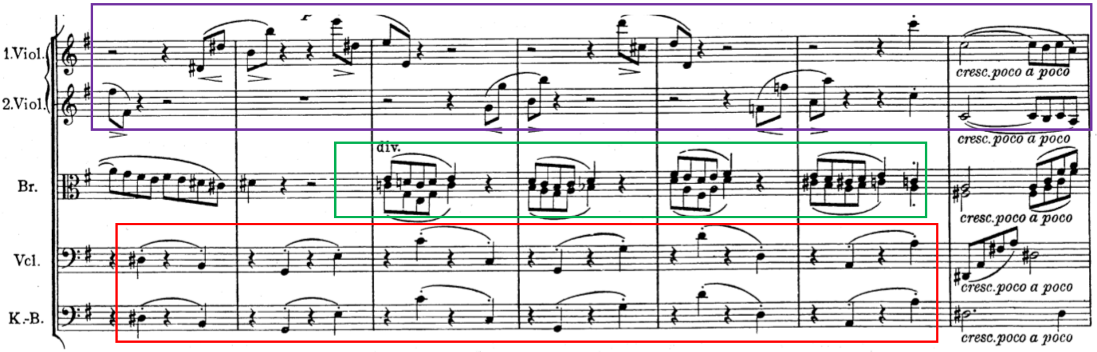
- Details of rhythm embeddings

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python

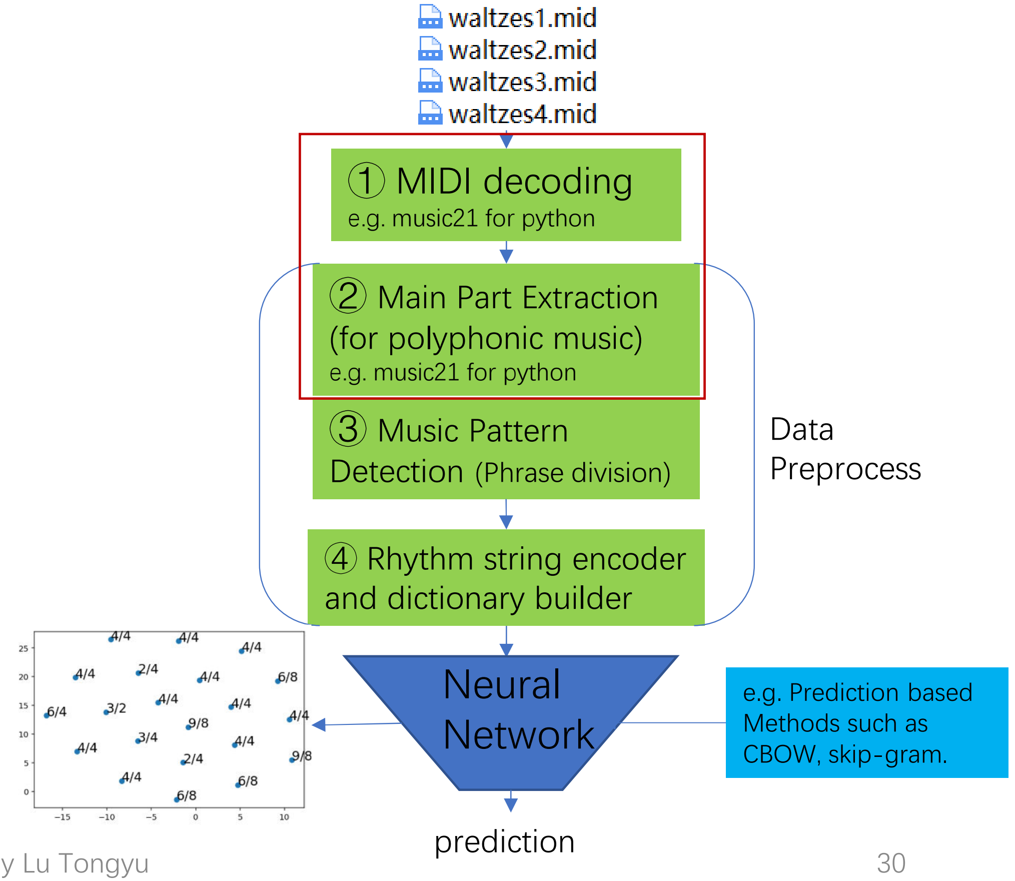
This task should be combined with music pattern detection if our music is polyphonic.

When to use main part extraction?
My proposal is as follows:

If there are several parts with different themes, we divide them into different parts.



Brahms: Symphony No.4, 1st movement



① MIDI decoding
e.g. music21 for python

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

prediction

Rhythm embedding: word2vec

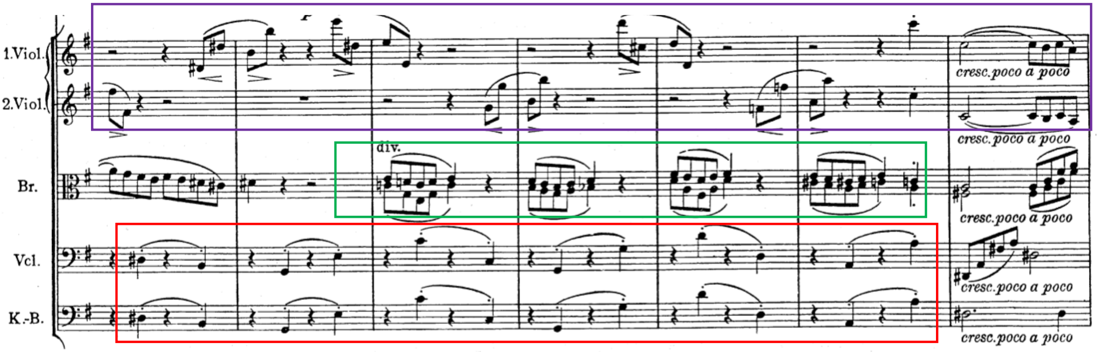
- Details of rhythm embeddings

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python

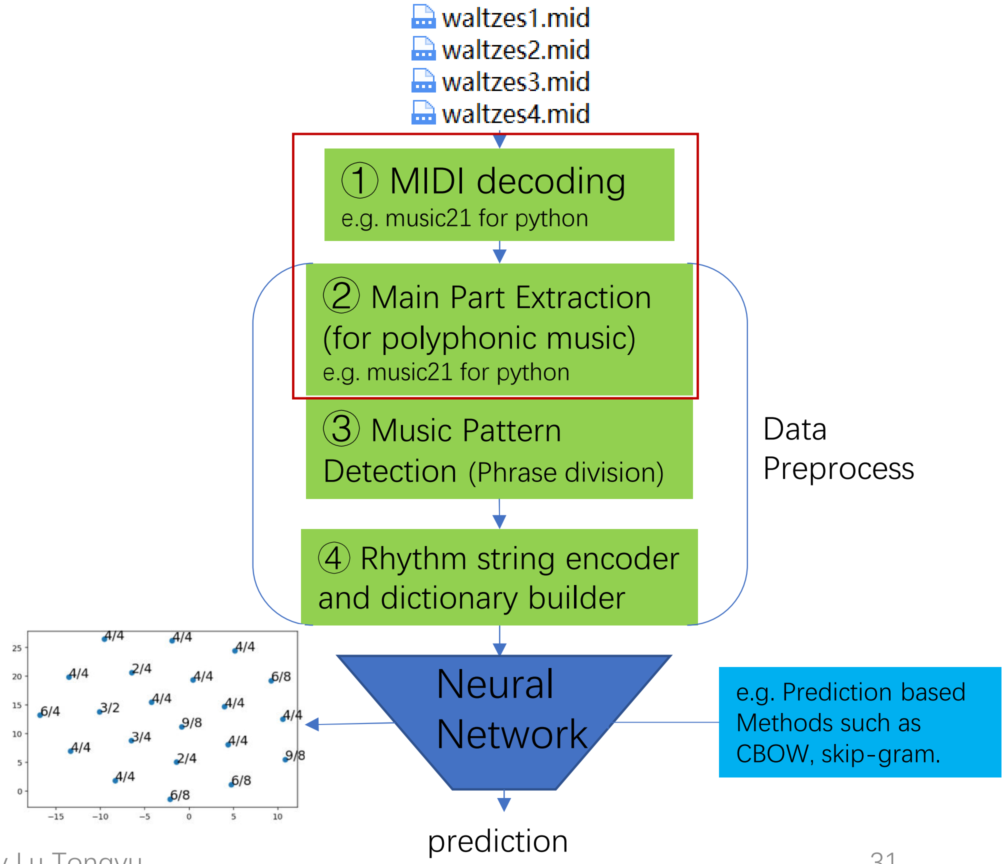
This task should be combined with music pattern detection if our music is polyphonic.

When to use main part extraction?
My proposal is as follows:

This task is out of range of our current concentration.



Brahms: Symphony No.4, 1st movement



① MIDI decoding
e.g. music21 for python

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

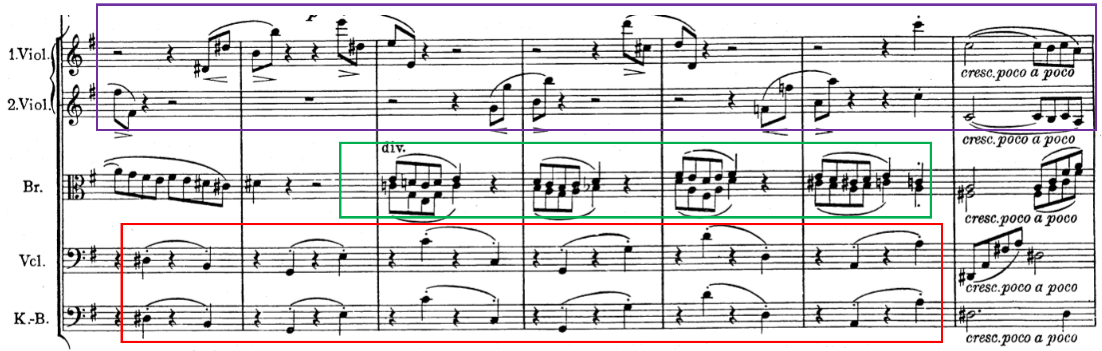
Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

prediction

Rhythm embedding: word2vec

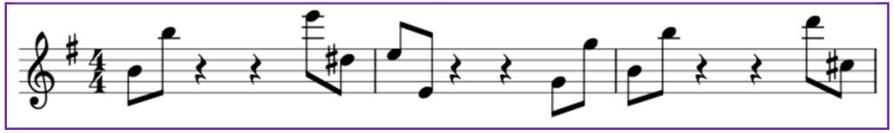
- Details of rhythm embeddings



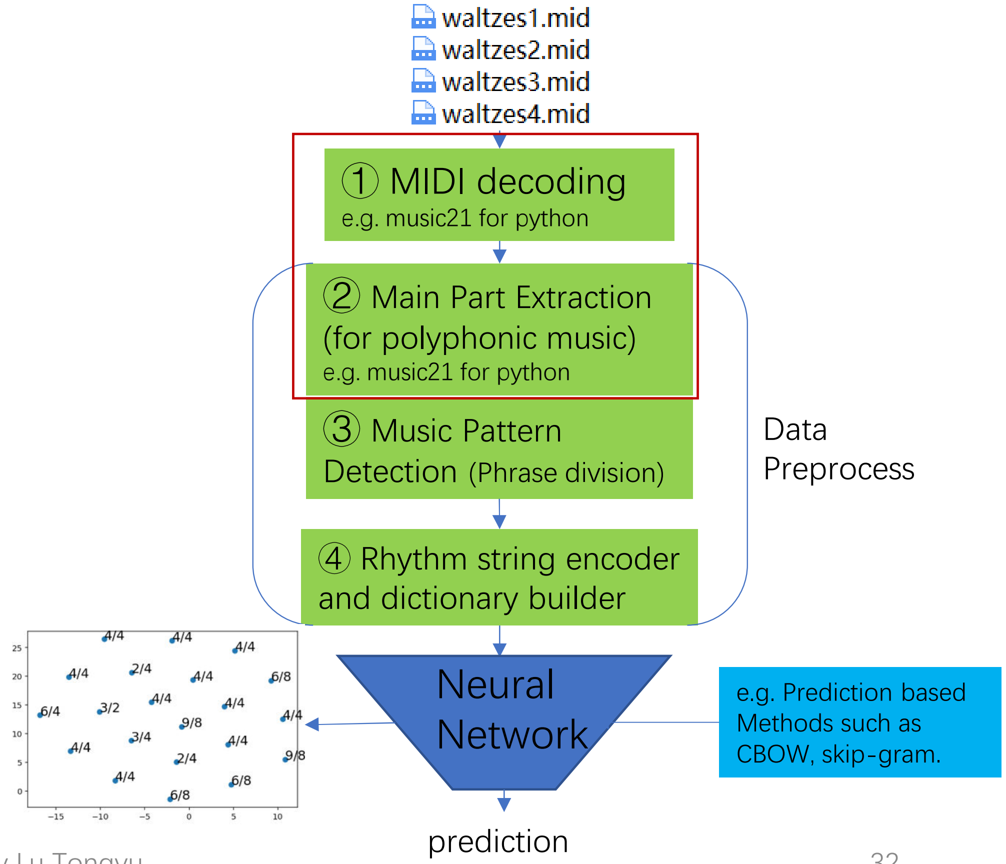
Format: e.g. music21 stream

Brahms: Symphony No.4, 1st movement

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python



Format: e.g. music21 stream



waltzes1.mid
waltzes2.mid
waltzes3.mid
waltzes4.mid

① MIDI decoding
e.g. music21 for python

② Main Part Extraction
(for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

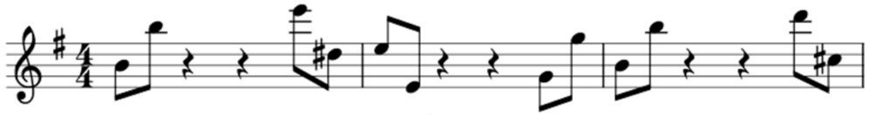
Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

prediction

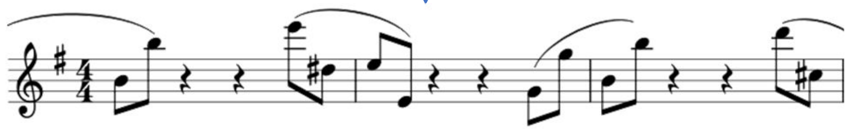
Rhythm embedding: word2vec

- Details of rhythm embeddings



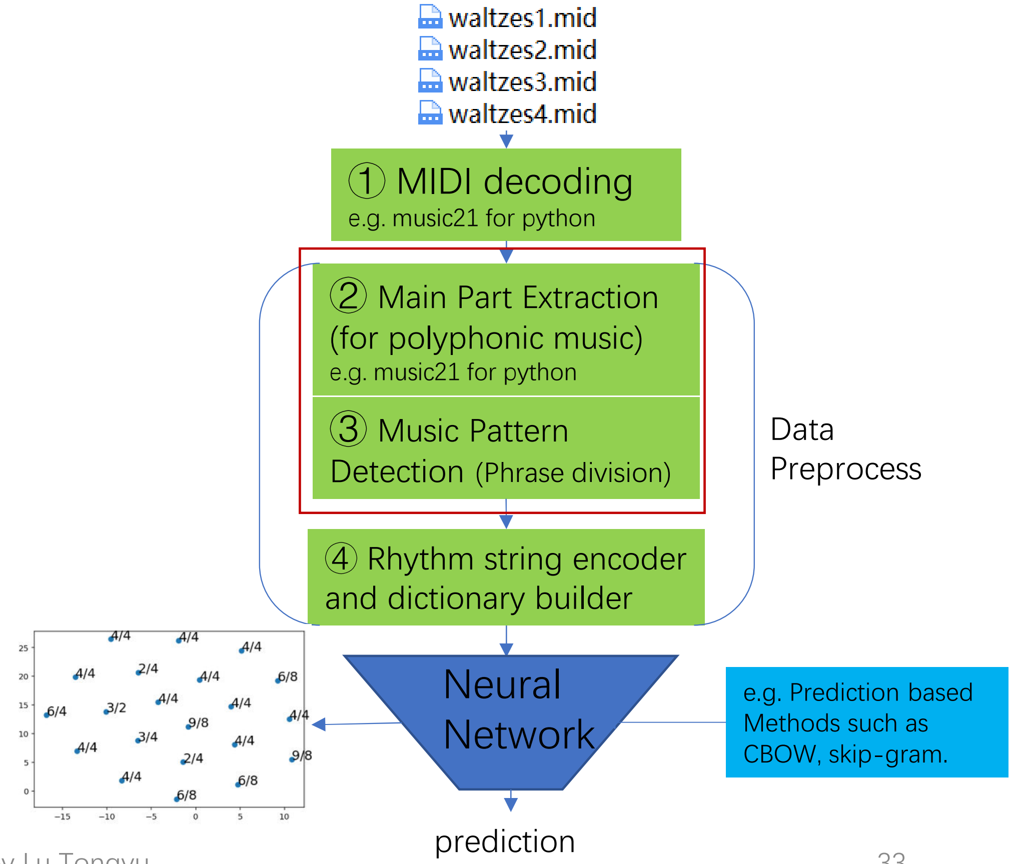
Format: e.g. music21 stream

③ Music Pattern Detection (Phrase division)



Format: e.g. music21 stream

(③ is optional, because we can still train rhythm embeddings without phrase marks. Still, if MIDI is able to label the breathes, we do not need to do this.)



waltzes1.mid
waltzes2.mid
waltzes3.mid
waltzes4.mid

① MIDI decoding
e.g. music21 for python

② Main Part Extraction (for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

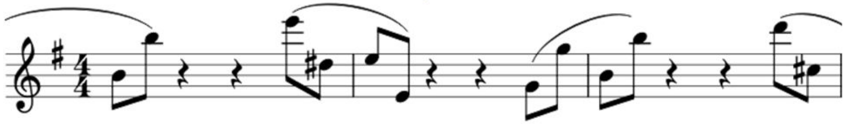
prediction

Rhythm embedding: word2vec

- Details of rhythm embeddings

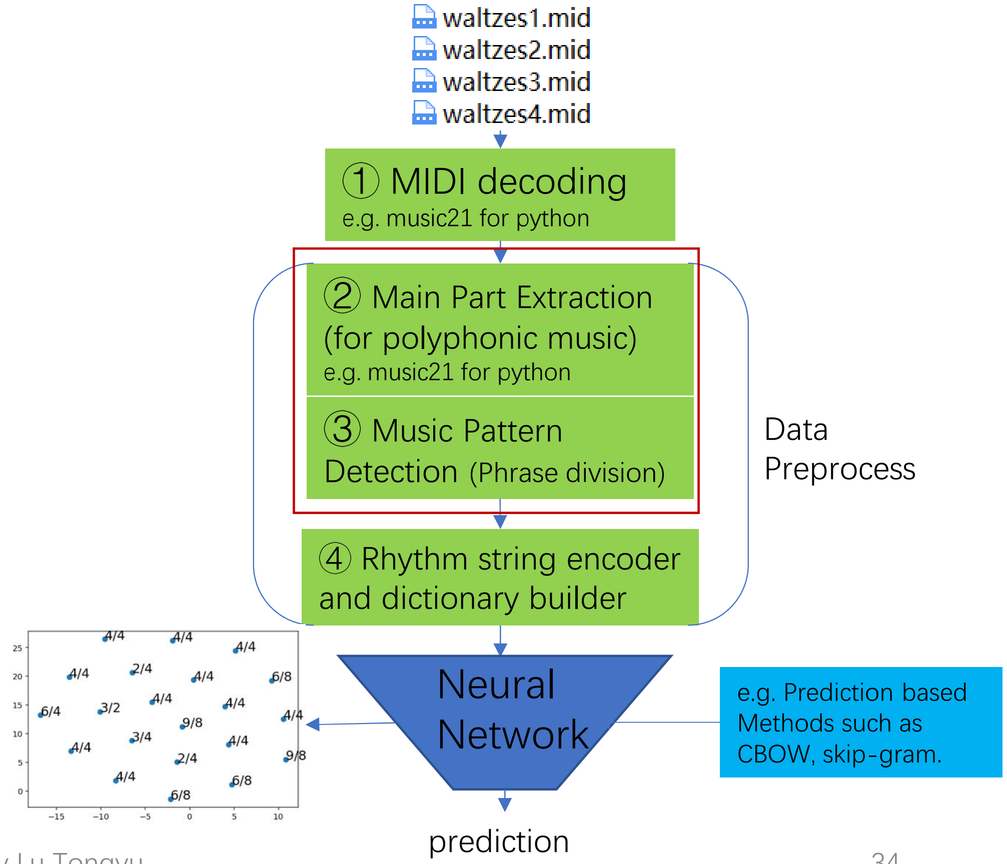
Format: e.g. music21 stream

③ Music Pattern Detection (Phrase division)



Format: e.g. music21 stream

Fortunately, the **Nottingham Music Database** is a medley of MIDI files of folk songs with normalized chord progressions.
Most folk song melodies are monophonic, so we do not need ②Main Part Extraction.



waltzes1.mid
waltzes2.mid
waltzes3.mid
waltzes4.mid

① MIDI decoding
e.g. music21 for python

② Main Part Extraction (for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

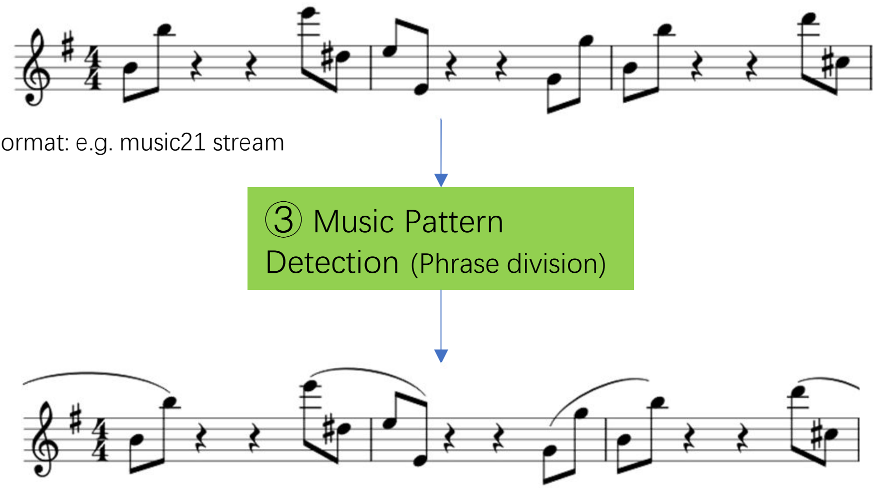
Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

prediction

Rhythm embedding: word2vec

- Details of rhythm embeddings

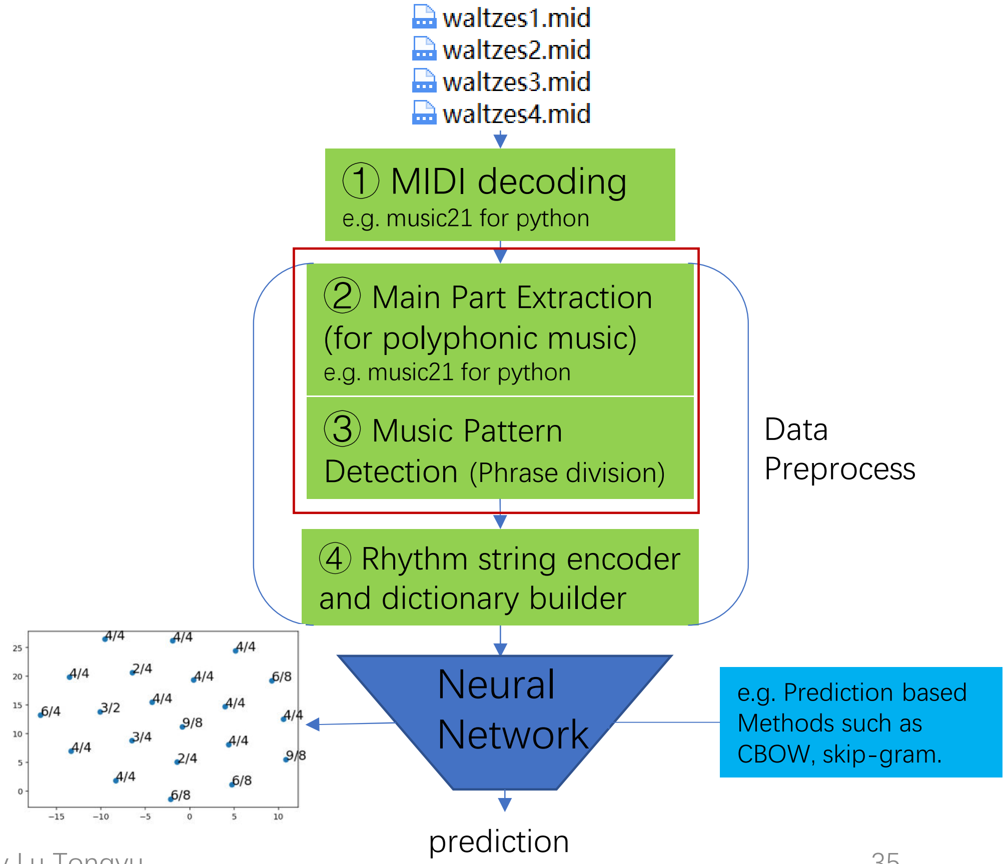


Format: e.g. music21 stream

③ Music Pattern Detection (Phrase division)

Format: e.g. music21 stream

However, **this database did not label phrases**, so we have to do phrase-hand-labeling for ③, or resort to the existing music pattern extraction algorithms (which are mostly undesirable in general cases).



waltzes1.mid
waltzes2.mid
waltzes3.mid
waltzes4.mid

① MIDI decoding
e.g. music21 for python

② Main Part Extraction (for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

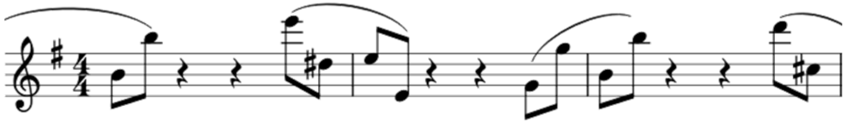
Neural Network

e.g. Prediction based Methods such as CBOW, skip-gram.

prediction

Rhythm embedding: word2vec

- Details of rhythm embeddings



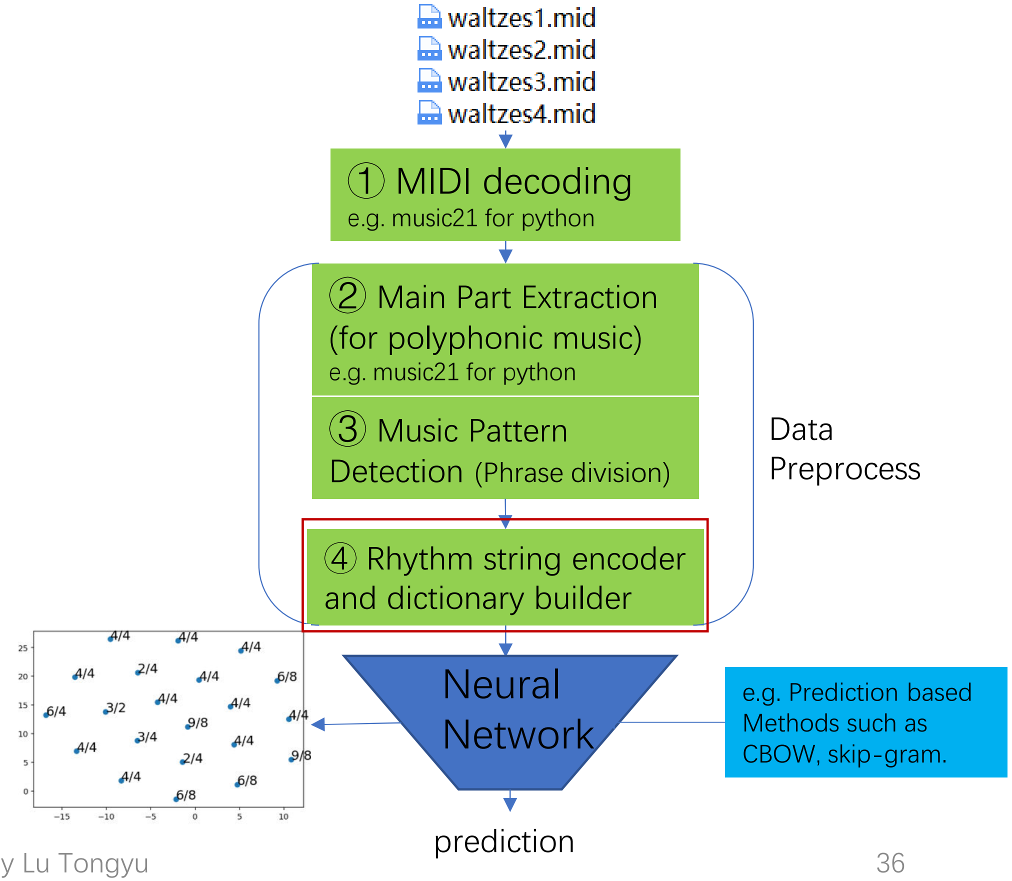
Format: e.g. music21 stream

④ Rhythm string encoder and dictionary builder

```
Rhythm_list = [0,2,5,5,5,1]

Rhythm_vocabulary = {0:"<BOS>", 1:"<EOS>", 2:"|4/4",
3:"<BREATH>", 4:"N0.500,N0.500,N1.000,N1.000,N0.500,N0.500|4/4",
5:"N0.500,N0.500,N1.000,N1.000,<BREATH>,N0.500,N0.500|4/4"}
```

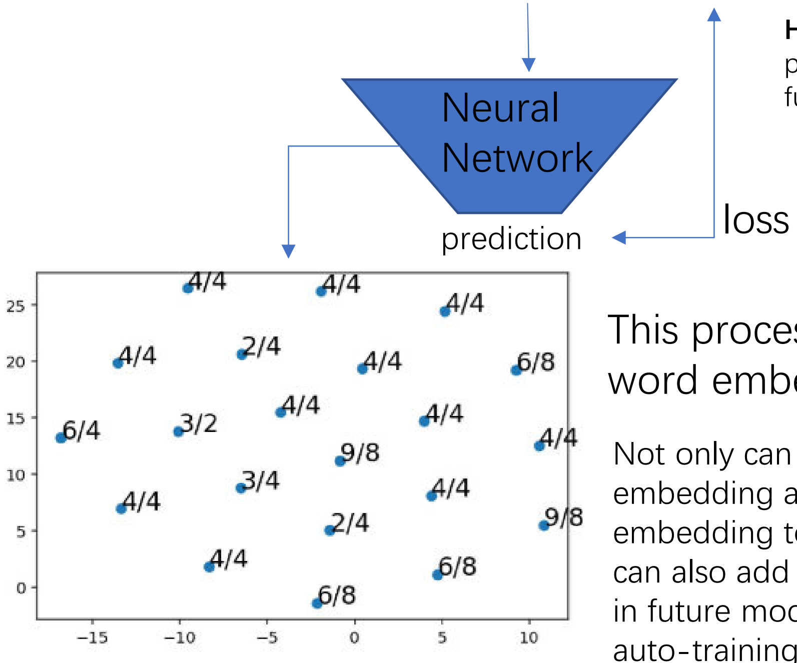
This process can be achieved by brute-force.



Rhythm embedding: word2vec

- Details of rhythm embeddings

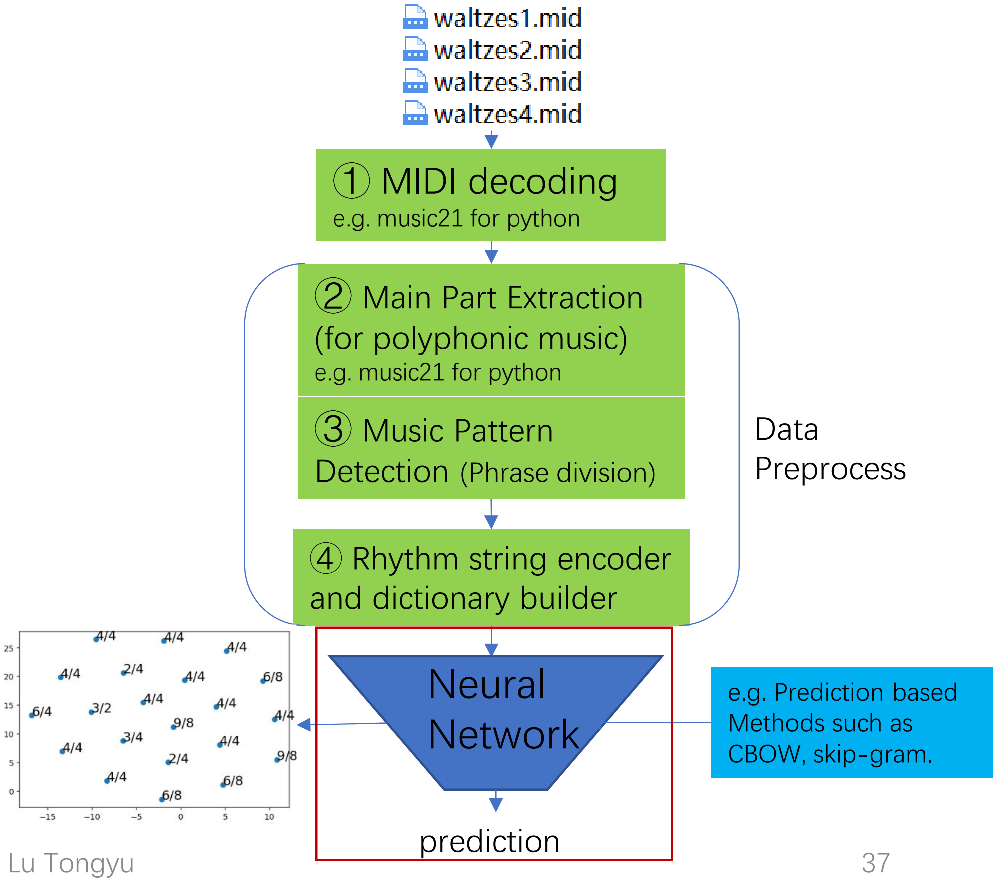
```
Rhythm_vocabulary = {0:“<BOS>”, 1:“<EOS>”, 2:“|4/4”,
3:“<BREATH>”, 4:“N0.500,N0.500,N1.000,N1.000,N0.500,N0.500|4/4”,
5:“N0.500,N0.500,N1.000,N1.000,<BREATH>,N0.500,N0.500|4/4”,…}
            Rhythm_list = [[0,2,5,5,5,1],…]
```



Hint: each sub-list is a piece. We may move further to polyphonic.

This process is similar to word embedding in NLP.

Not only can we pre-train the embedding and feed the embedding to future models, we can also add embedding layers in future models to achieve auto-training.



① MIDI decoding
e.g. music21 for python

② Main Part Extraction (for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

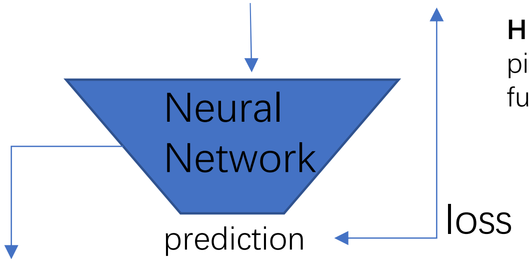
Data Preprocess

e.g. Prediction based Methods such as CBOW, skip-gram.

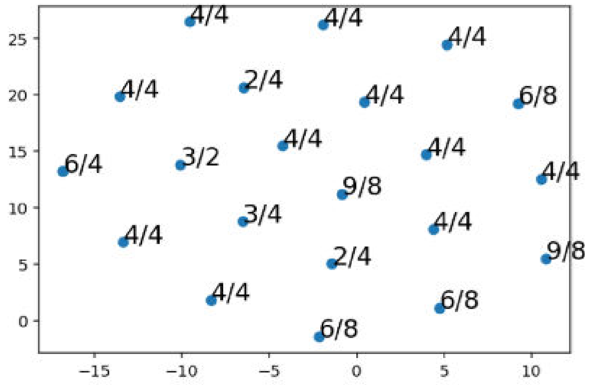
Rhythm embedding: word2vec

- Details of rhythm embeddings

```
Rhythm_vocabulary = {0:“<BOS>”, 1:“<EOS>”, 2:“|4/4”,
3:“<BREATH>”, 4:“N0.500,N0.500,N1.000,N1.000,N0.500,N0.500|4/4”,
5:“N0.500,N0.500,N1.000,N1.000,<BREATH>,N0.500,N0.500|4/4”,…}
                Rhythm_list = [[0,2,5,5,5,1],…]
```

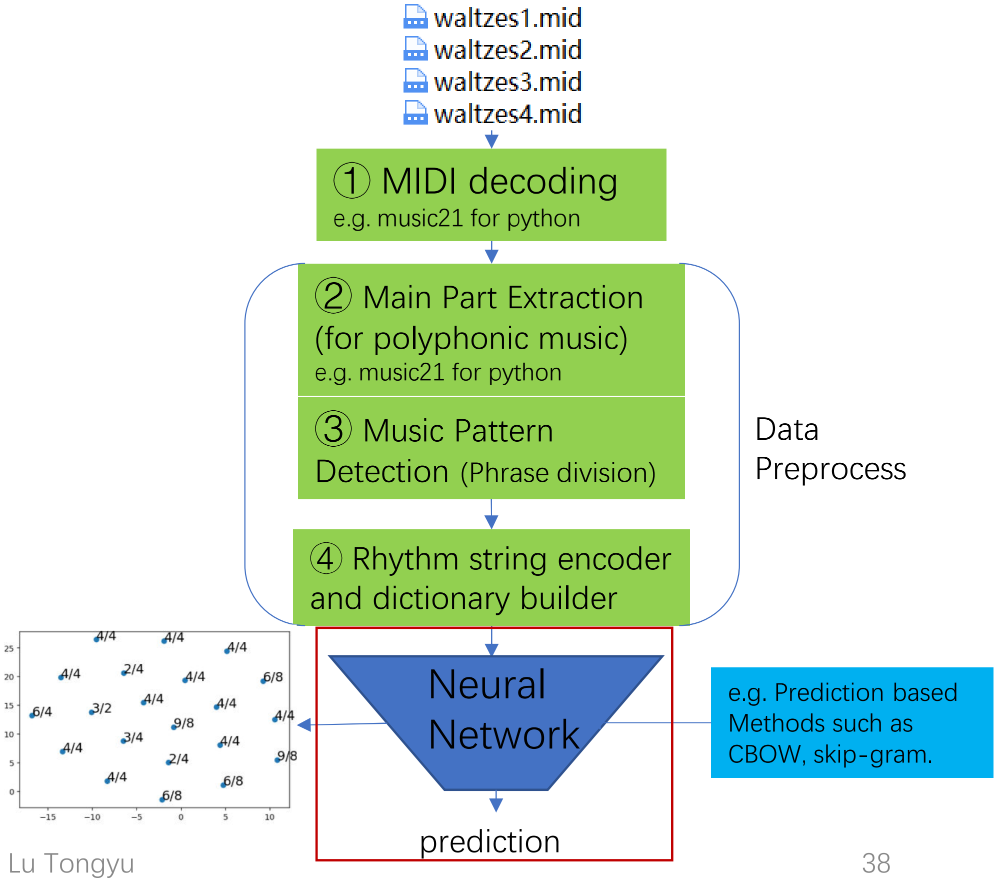


Hint: each sub-list is a piece. We may move further to polyphonic.



By the way, this scheme **devolves the task of phrase division** to the **task of understanding rhythm word embedding context**.

A new idea of music pattern detection!



① MIDI decoding
e.g. music21 for python

② Main Part Extraction (for polyphonic music)
e.g. music21 for python

③ Music Pattern Detection (Phrase division)

④ Rhythm string encoder and dictionary builder

Data Preprocess

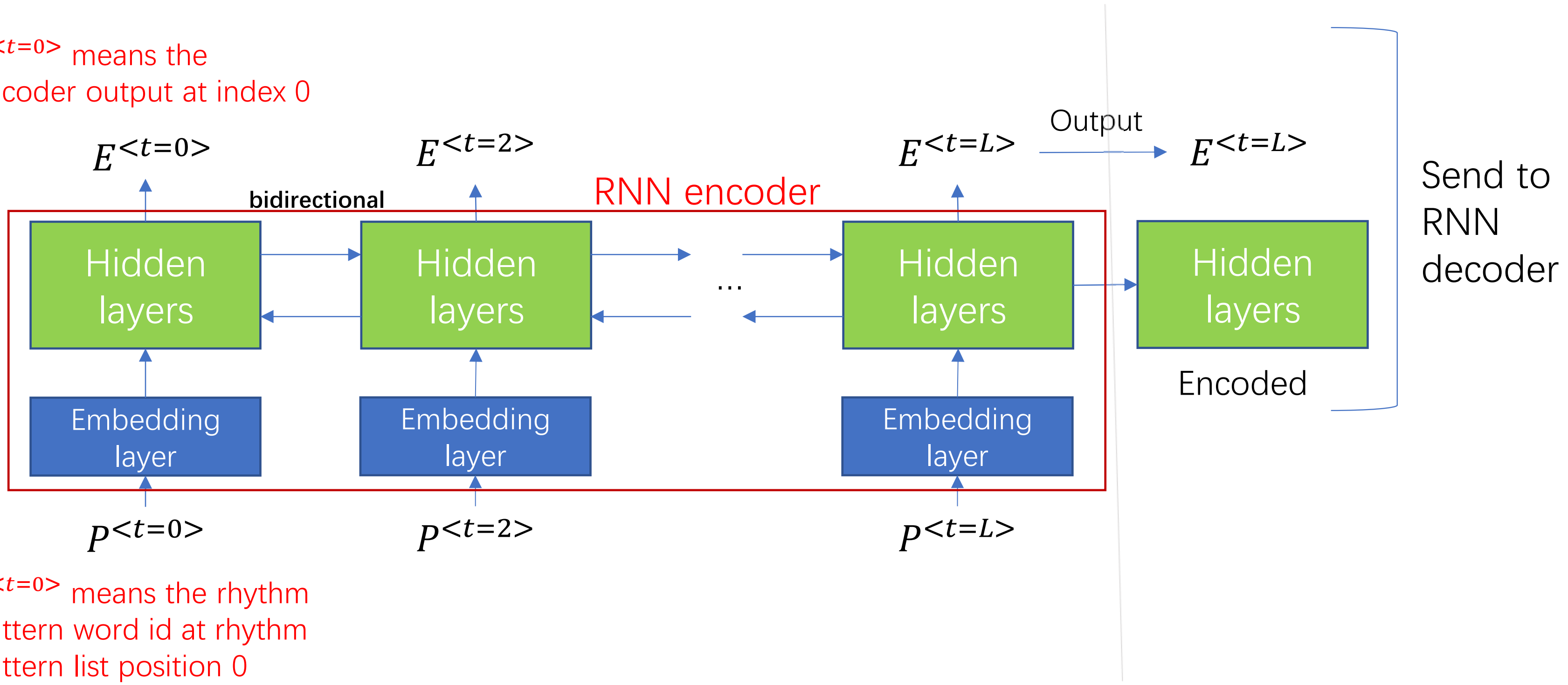
e.g. Prediction based Methods such as CBOW, skip-gram.

Machine Learning Models for Generation

- Baseline: seq2seq model without attention

A seq2seq model is essentially a RNN auto-encoder.

$E^{<t=0>}$ means the encoder output at index 0



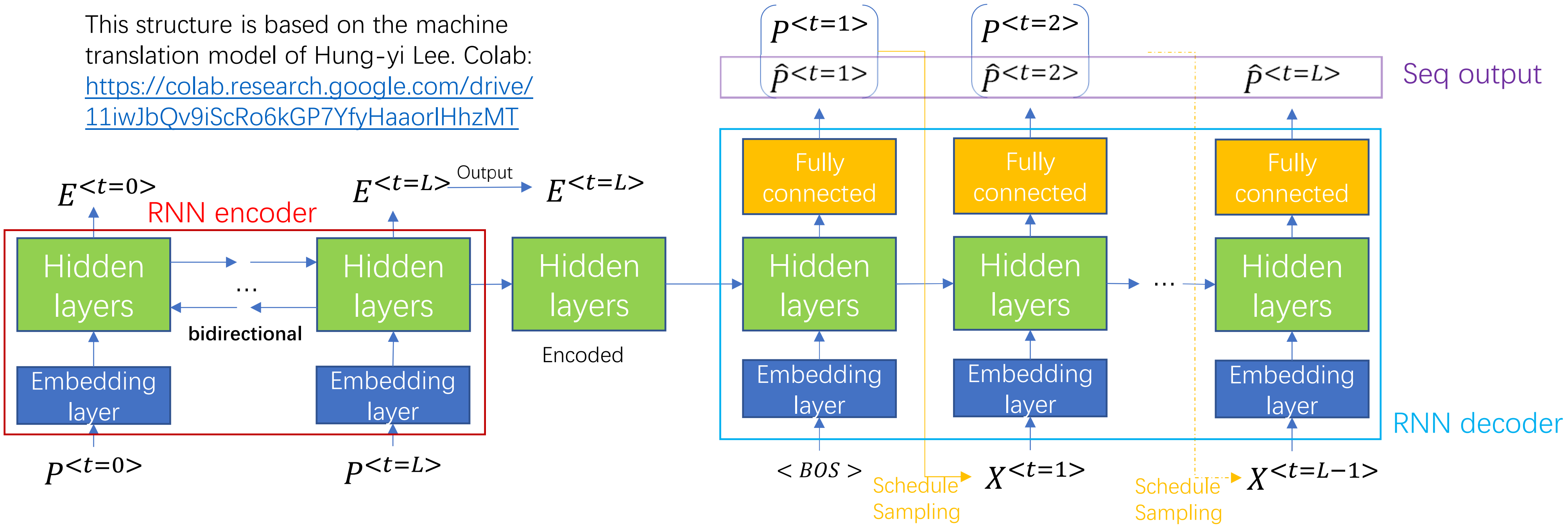
$P^{<t=0>}$ means the rhythm pattern word id at rhythm pattern list position 0

Machine Learning Models for Generation

- Baseline: seq2seq model without attention

A seq2seq model is essentially a RNN auto-encoder.

This structure is based on the machine translation model of Hung-yi Lee. Colab: https://colab.research.google.com/drive/11iwJbQv9iScRo6kGP7YfyHaaorIHhzMT



$X^{<t=1>}$ means the decoder input at time 1

Machine Learning Models for Generation

- Baseline: seq2seq model without attention

Experiment results…

baseline_the_test_stream_result0.mid
(meter in 6/8)

baseline_the_test_stream_result1.mid
(meter in 6/8)

training_data0
(meter in 4/4)

The baseline outputs do generate phrases.

However, the phrases are not in normalized modes (e.g. the mode of 4 meters per phrase).

Moreover, the endings of baseline outputs are undesirable.

Colab for data preprocess (temp, editable, produced by Lu, referred to Yan’ s naïve version):
https://colab.research.google.com/drive/1IDN2LmHovC40L1X7jpNmLcDhNlkmwGxx?usp=sharing
Colab for baseline seq2seq model (temp, editable, produced by Lu, referred to Hung-yi Lee’ s HW):
https://colab.research.google.com/drive/1Fi3e-RuxcbK-7KoSLunHiXuh2fg-bC7c

Machine Learning Models for Generation

- Problem during experiment

· The model tends to repeat the same word over and over again

This phenomenon is called "neural text degeneration".

A. Holtzman, H. Buys, Li Du et.al. *(2020).* *"The Curious Case of Neural Text Degeneration"* *(PDF).* ICLR 2020

Naïve solutions: schedule sampling, beam search.

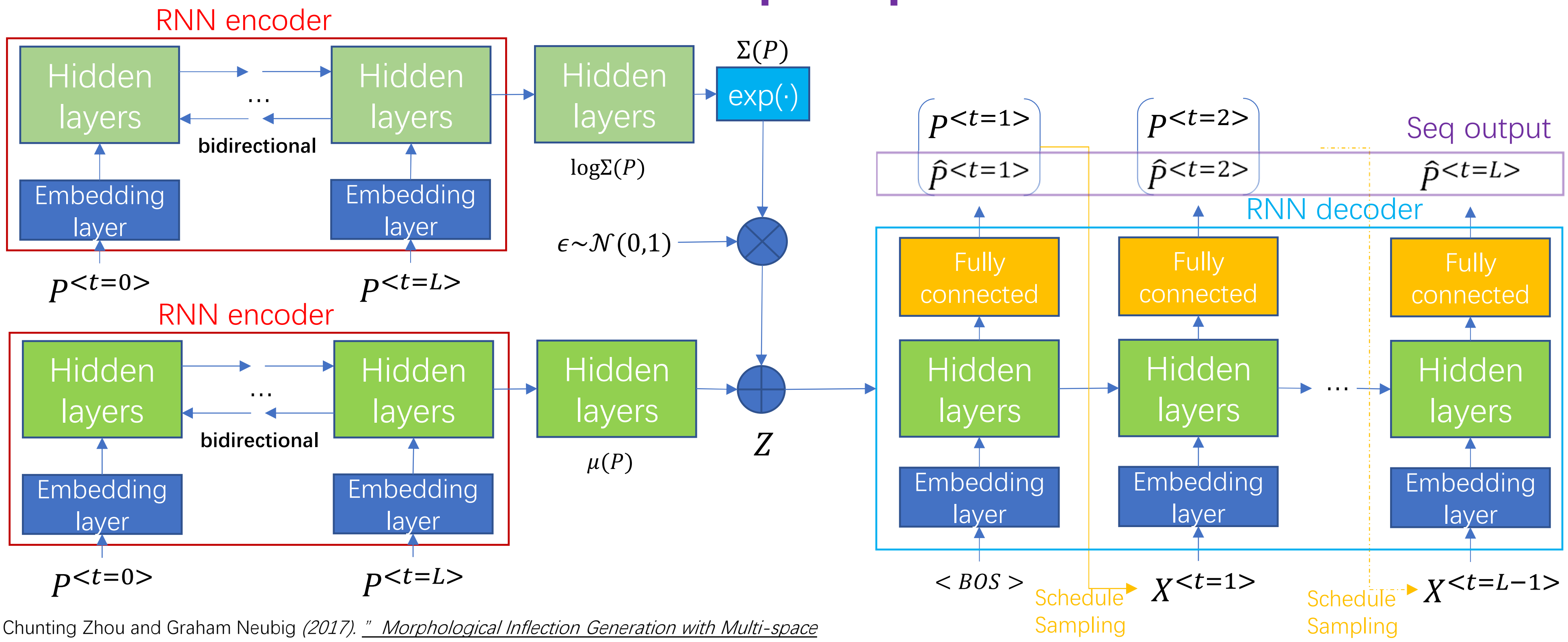
Questions: are there other methods to deal with this problem?

Speculations:

1. The model forgot the previous information. Then, at a time, it only knows the past few elements, which are repeating elements.
2. Parameters found a local minima, which is much easier to reach than getting the global maxima.
3. The window of word embedding is too small.

Machine Learning Models for Generation

- TODO #1: variational seq2seq models

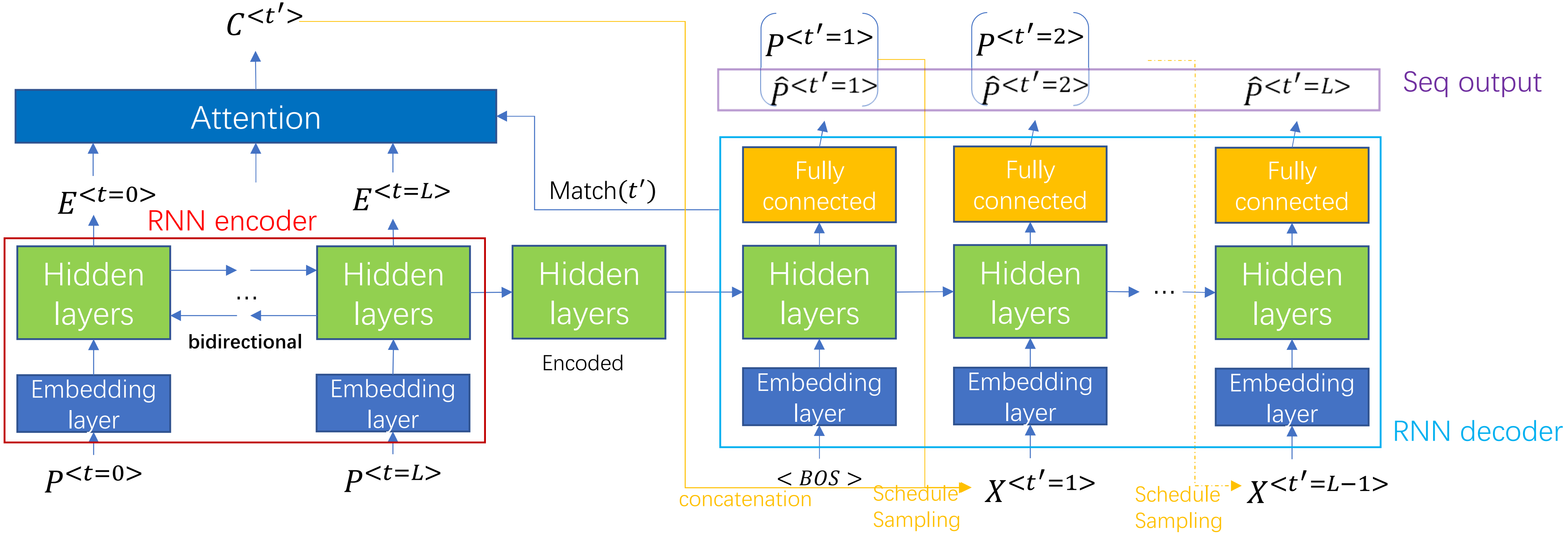


Chunting Zhou and Graham Neubig *(2017).* *" Morphological Inflection Generation with Multi-space Variational Encoder-Decoders"* *(PDF).* Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection, pages 58–65, Vancouver, Canada, August 3–4, 2017

Machine Learning Models for Generation

- TODO #2: seq2seq+attention

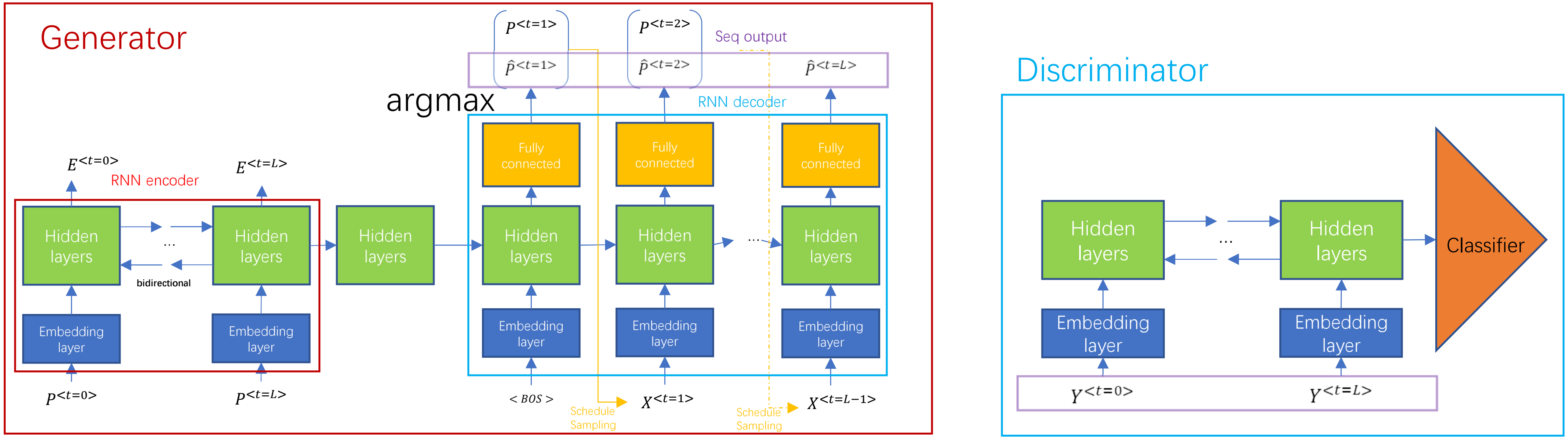
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin*(2017). "*
Attention is All You Need*"* *(PDF).* https://arxiv.org/abs/1706.03762 31st Conference on Neural Information Processing Systems



Hung-yi Lee, Slides, http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2017/Lecture/Attain%20(v5).pdf

Machine Learning Models for Generation

- TODO #3: seq2seq+GAN



Hints:

This GAN structure is somehow like C-RNN-GAN.

Olof Mogren(2016). "C-RNN-GAN: Continuous recurrent neural networks with adversarial training" (PDF).
Constructive Machine Learning Workshop (NIPS 2016), Barcelona

Machine Learning Models for Generation

- Problems during experiment of seq2seq GAN

· The backpropagation from discriminator to generator may face non-differentiable argmax operation.

Candidate solutions:

1. Modify the structure of discriminator: delete the embedding layer and use distribution vectors as outputs of the decoder.
2. Use the Gumbel-Softmax trick.
3. Modify the model to ForGAN structures.

Alireza Koochali, Peter Schichtel, Sheraz Ahmed, Andreas Dengel*(2016). "Probabilistic Forecasting of Sensory Data with Generative Adversarial Networks – ForGAN" arXiv:1903.12549v1*

Moreover, it is worthwhile to try WGAN.

Machine Learning Models for Generation

- TODO #4: build embedding layers which take account of rhythmic edit distances
- TODO #5: beam search
- TODO #6: Gumble-Softmax
- TODO #7: WGAN
- TODO #8: GAN+VAE+Attention+RNN?
- Any other suggestions?

More to Consider

- How to evaluate experimental results?

  BLEU score for reconstruction evaluation.

- How to do actual generation rather than reconstructing?

  Throw away the encoder and feed noisy hidden states to decoder?
  Combine a medley of hidden states generated by different training data?

- How to evaluate the actual generations of model?

  Turing test?

An Overall TODO List

To do in recent future:

- Add phrase labels to training data and utilize them
- Rhythm embedding considering note-level information
- Try various network architectures: variational seq2seq, seq2seq GAN, WGAN
- Try various tricks: beam search, Gumble-Softmax
- From reconstruction to actual generation
- Evaluation of results

To do further:

- Consider word embedding representation of chords
- Combine rhythm with chords, and construct a multidimensional musical word embedding
- Adopt main part extraction algorithms for data preprocessing
- Explore better machine learning structures

Thank You For Watching