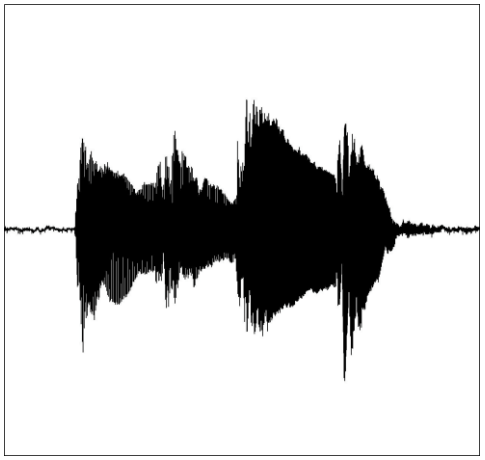
A Study on LSTM Networks for Polyphonic Music Sequence Modelling

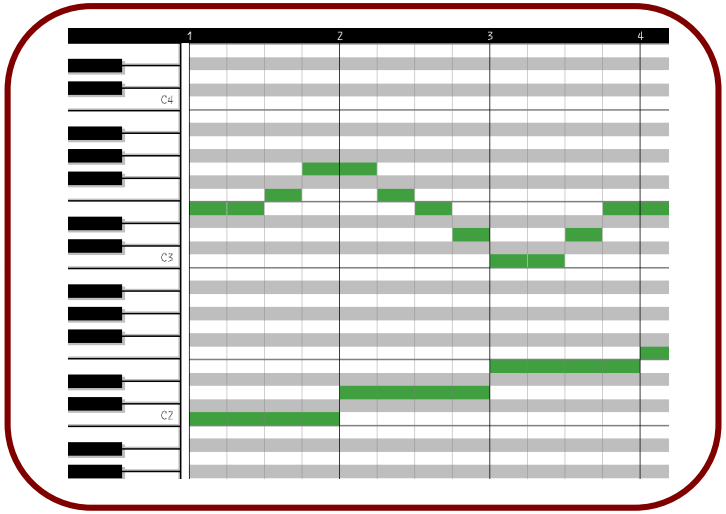
Adrien Ycart, Emmanouil Benetos

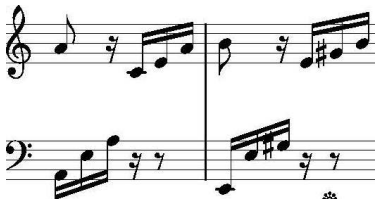
Published in: Proceedings of the 17th International Society for Music Information Retrieval Conference

Centre for Intelligent Sensing
Queen Mary University of London

Automatic Music Transcription (AMT)

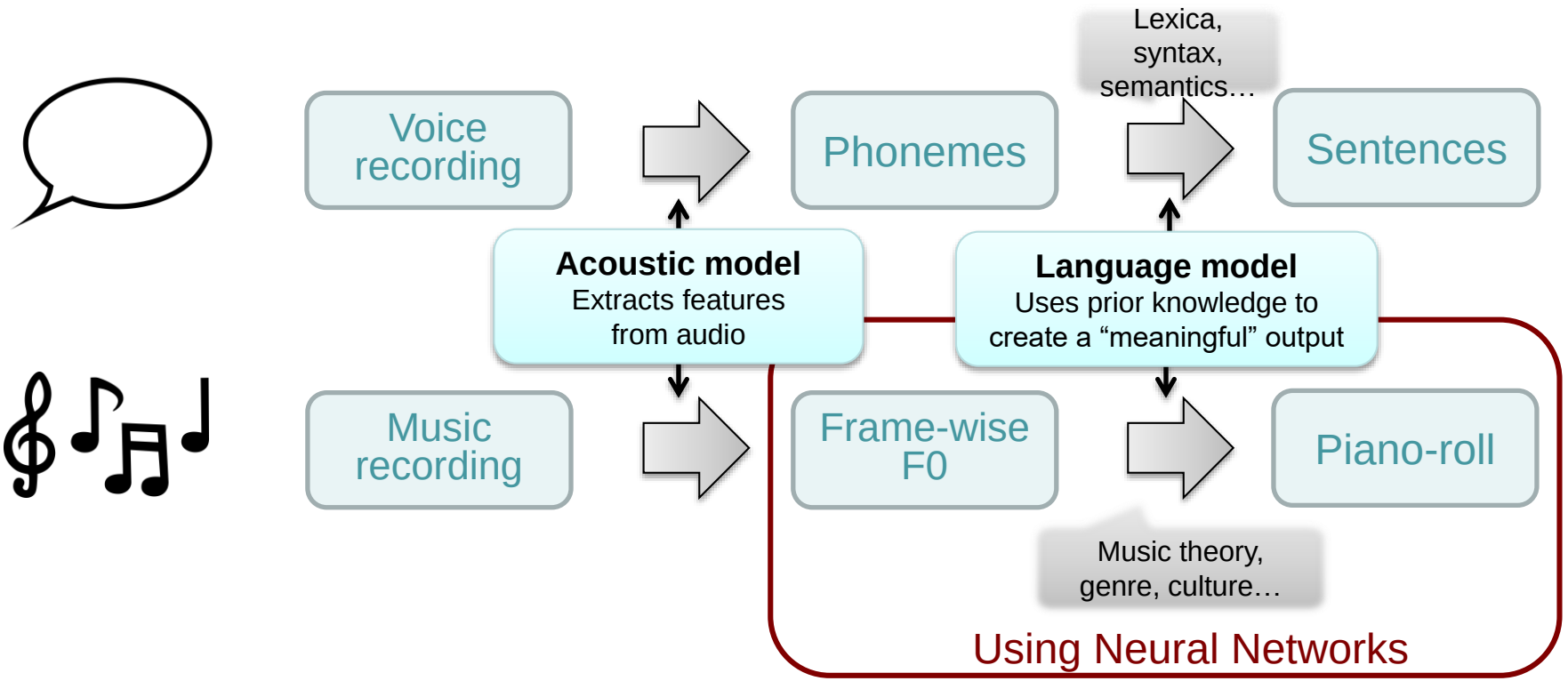






C	F	G	Dm
G	F	Em	C

Music Language Models



State of the art

- Boulanger-Lewandowski et al. (2012):
  - Symbolic music modelling
  - RNN-RBM architecture

    → **Time step = a sixteenth-note**
- Sigtia et al. (2015):
  - Integrates the RNN-RBM language model with a variety of neural acoustic models

    → **Time step = 10ms !!**
- Problem:
  - Time step too short compared to the typical length of a note
  - No normalisation with respect to the tempo

Our aim

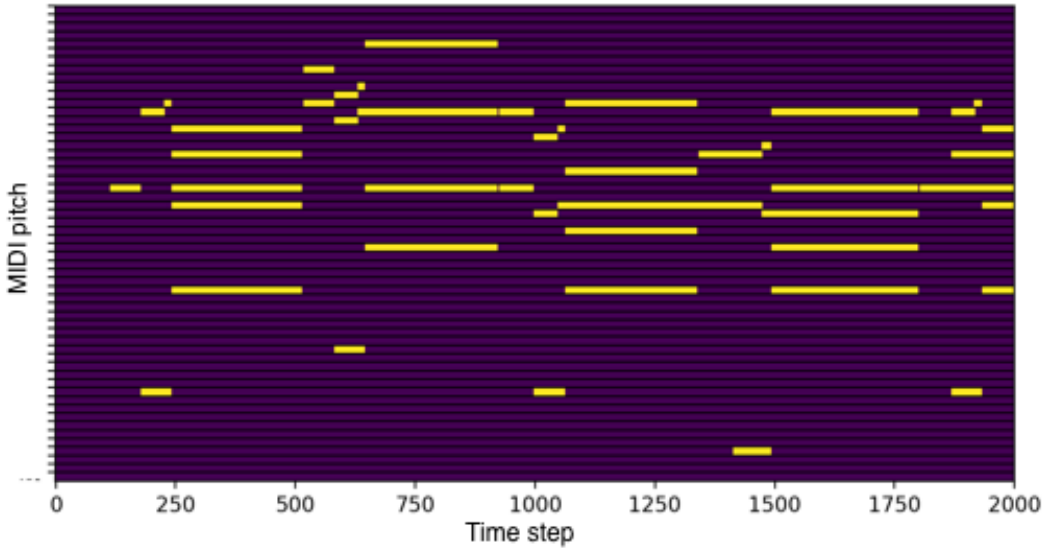
- Start with a **simple** architecture : single-layer LSTM
- Built incrementally using an **experimental method**
- Make **musically-motivated** choices
- Evaluate on a **prediction** task the influence of:
  - Number of hidden nodes
  - Learning rate
  - Time-step
  - Data augmentation

Data representation

- Piano-roll:
  - 88 x T matrix, where T is the number of time steps
  - Binarized : $M[i,j] = 1$ iff pitch i is active at time step j
  - No distinction between onset and continuation

→ Given time steps $0, 1, \ldots t\text{-}1$, predict time step t

- Two time steps:
  - Time-based: 10ms
  - Note-based: a sixteenth-note



Network Architecture

- Single-layer LSTM
  - 88 inputs
  - One hidden layer
  - 88 outputs
- Hidden nodes $\in \{64, 128, 256, 512\}$
- Adam optimiser, cost function: cross-entropy
- Learning rate $\in \{0.01, 0.001, 0.0001\}$
- No Dropout

Output → Sigmoid → Threshold → Binary prediction

→ Compare to ground truth : precision, recall, F-measure

Datasets

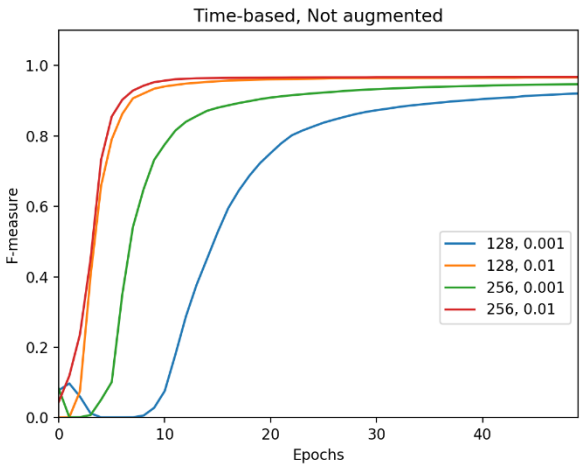
- *Synth* dataset :
  - Synthetic data
  - Only notes in C major scale
  - Sequences of 3 chords, allow repetition
  - Each chord has 1, 2 or 3 notes
  - 1 second each (=1 quarter-note)
  - Overall: 36000 files, 30 hours of data
- *Piano* dataset:
  - Real-world data
  - 307 classical piano pieces
  - Natural interpretation
  - Contains the rhythmic ground truth
  - Only keep the 1st minute of each file
  - Overall: 5 hours, 60 hours with data augmentation

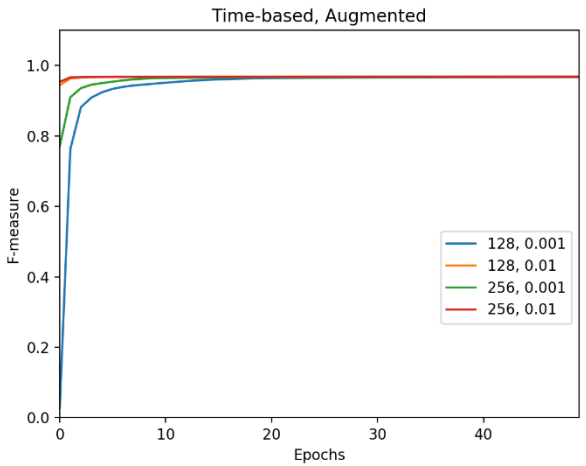
Hidden nodes, learning rate, data augmentation

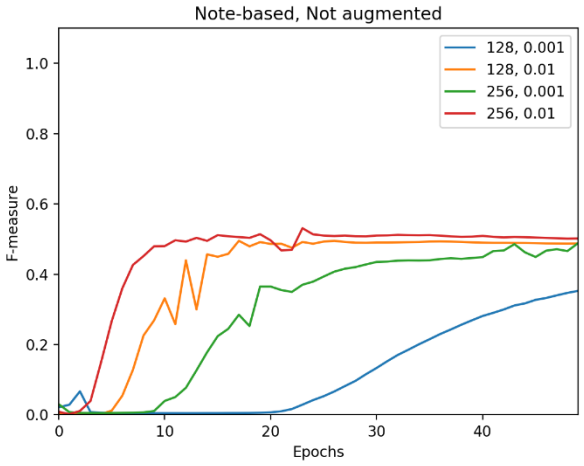
- Results:
  - Trained and tested on real data

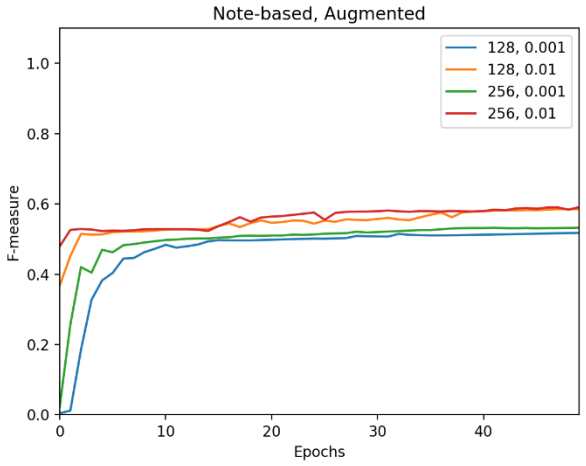
	F-Measure	Precision	Recall
Without augmentation			
128, 0.001	0.451	0.409	0.509
256, 0.001	0.513	0.484	0.549
128, 0.01	0.548	0.536	0.560
256, 0.01	**0.553**	**0.544**	**0.562**
With augmentation			
128, 0.001	0.558	0.557	0.560
256, 0.001	0.571	0.552	0.592
128, 0.01	0.597	0.61	0.588
256, 0.01	**0.601**	**0.615**	**0.592**

Results on test dataset for note-based time-steps



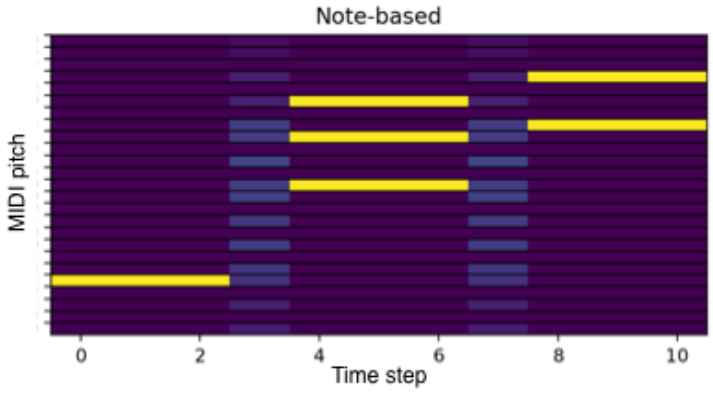




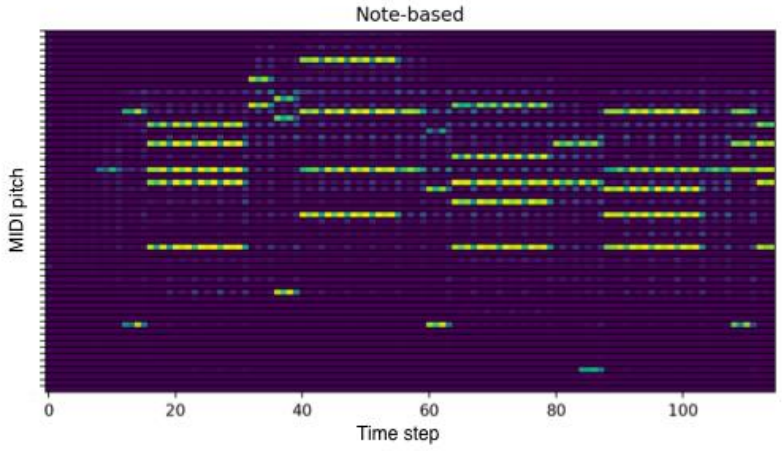


Time steps

- Better prediction performance with time-based time steps:
  - Time-based: F-measure = 0.96
  - Note-based: F-measure ~ 0.60
- Time-based: mostly self-transitions → **simple smoothing !**
- Note-based: more interesting **musical** properties



Synthetic data



Real data

Preliminary experiment on audio transcription

- Post-processed the output of an acoustic model (*Benetos and Weyde, 2015*) with our system
- Tried both on raw posteriogram, and on thresholded posteriogram
- Tried with models trained on synthetic data, on real data, with time-based and note-based timesteps
- Tried only on one piece

Preliminary experiment on audio transcription

- Results:
  - Mostly, the results were worse than the acoustic model alone
  - Only with time-based time steps they were slightly better
    - →Simple smoothing
  - Trained on synthetic data : masks out out-of-key notes

	F-Measure	Precision	Recall
Full audio, raw_piano			
Baseline	0.455	0.960	0.299
128, 0.001	0.458	0.938	0.303
256, 0.001	0.458	0.941	0.303
128, 0.01	0.460	0.959	0.303
256, 0.01	**0.460**	**0.961**	**0.303**
Right hand in C, raw_post, Synth			
Baseline	**0.670**	0.898	**0.535**
128, 0.001	0.556	0.955	0.393
256, 0.001	0.607	**0.966**	0.442
128, 0.01	0.522	0.834	0.380
256, 0.01	0.527	0.877	0.377
Full note-based, raw_piano			
Baseline	**0.526**	**0.963**	**0.361**
128, 0.001	0.434	0.624	0.332
256, 0.001	0.440	0.651	0.332
128, 0.01	0.478	0.852	0.332
256, 0.01	0.481	0.875	0.332

Conclusion

- Evaluation : better prediction F0 ≠ better modelling !
  - Have to find a more relevant way to evaluate
  - Evalutate cross-entropy and F0 only on transitions (ie the difficult cases)
- Perspectives:
  - More experiments:
    - More hidden layers
    - Inspect their activations
  - Integrate our language model with an acoustic model to transcribe music from audio
    - Note-based time steps : will require beat tracking

Thank you