Lyrics to Melody project report

Jaydan Aladro[1,2] **Maxime Bélanger**[2] **Guillaume Charron**[1,2]
Jeremy Kaufman[1,2]

[1]Mila - Institut québécois d'intelligence artificielle
[2]Université de Montréal

Cet article explore l'efficacité des modèles génératifs pour la génération de mélodies au format MIDI à partir de paroles de chansons. Les fichiers MIDI, en encodant des informations musicales telles que la hauteur et la durée, permettent une interprétation et une reproduction précises des mélodies par les ordinateurs. Pour les artistes, cette approche simplifie le processus d'écriture de chansons, notamment lorsqu'elle est combinée à des paroles préexistantes. Plusieurs modèles d'IA, incluant les réseaux LSTM, les GANs et les mécanismes d'attention, ont été utilisés pour générer des mélodies à partir de paroles. Notre approche, basée initialement sur le modèle transformateur T5 puis sur une architecture RNN, vise à résoudre les limitations rencontrées dans la génération de mélodies variées. Pour améliorer les performances et réduire le surapprentissage, nous introduisons des techniques d'augmentation de données telles que la segmentation par fenêtre glissante et le décalage de hauteur de notes. Les mélodies résultantes, générées dans des fichiers MIDI puis converties en fichiers audio, présentent une excellente cohérence et diversité.

Lien vers le code: Lyrics2Melody

1 Introduction

Ces dernières années, la modélisation générative a révolutionné la génération d'images et de textes, mais la création de mélodies a été moins explorée. Lors du processus créatif initiale d'une chanson, la mélodie et les paroles sont fortement corrélés, surtout au niveau des syllables, telles que démontré dans de nombreuses études comme celle de Nichols et al. (2009). Il est donc très utile pour les artistes de pouvoir générer des mélodies à partir de paroles ou d'idées de paroles, en leur permettant de concentrer leur créativité sur celles-ci et les accompagnements au lieu d'aussi avoir à créer des mélodies variées.

Plusieurs modèles d'intelligence artificielle ont été développés pour relever ce défi, chacun apportant des perspectives et méthodologies uniques. Par exemple, le travail de Yu et al. (2021) sur les LSTM-GANs conditionnels pour la génération de mélodies à partir de paroles a démontré l'efficacité de cette approche pour produire des mélodies réalistes. De manière similaire, le réseau d'alignement basé sur l'attention pour la génération de mélodies à partir de paroles incomplètes, tel qu'étudié par M et al. (2023) a exploré avec succès la prédiction de paroles et de mélodies même avec des entrées lyriques incomplètes, en utilisant les mécanismes d'attention et un réseau de neurones récurrent basé sur les LSTMs.

Notre recherche a initialement utilisé un modèle transformateur T5, mais face aux limitations rencontrées telles des notes répétitives et du sur-apprentissage, nous avons

adopté une architecture RNN simplifiée. Pour améliorer la performance et réduire le sur-apprentissage, nous avons également mis en œuvre des techniques d'augmentation de données, comme la segmentation par fenêtre glissante et le décalage de hauteur. En utilisant les probabilités générées par notre modèle, nous avons réussi à générer des mélodies cohérentes, structurées et diversifiées. Ce travail contribue ainsi à l'évolution de la génération de mélodies à partir de paroles, en proposant des architectures de modèles plus efficaces et des stratégies d'entraînement et d'augmentation de données innovantes.

2 Revue de la littérature

Le travail de Yu et al. (2021) utilise des réseaux LSTM-GAN conditionnels pour créer des mélodies basées sur des paroles, combinant les capacités des LSTM avec le pouvoir discriminatif des GAN pour produire des mélodies réalistes. Ils ont créé un ensemble de données de plus de 12 000 séquences de 20 notes, démontrant la capacité de leur modèle à générer des mélodies pertinentes par rapport aux paroles. La technique nécessite une quantisation des valeurs continues pour aligner les mélodies agréablement.

En revanche, l'article M et al. (2023) aborde la génération de mélodies à partir de paroles incomplètes en utilisant un réseau d'alignement basé sur l'attention. Ce modèle capte la relation sémantique entre les paroles et les mélodies, fonctionnant bien même avec des entrées textuelles partielles. Ils proposent une méthode novatrice qui inclut un modèle lyric2vec pour améliorer l'alignement des paroles et des mélodies.

3 Méthodes

Cette section décrit en détail les méthodes et données utilisées dans notre étude pour la génération de mélodies à partir de paroles. Nous avons exploré deux approches principales : l'utilisation d'un modèle transformer T5 (Raffel et al., 2023) avec une architecture encodeur-décodeur personnalisée et l'utilisation d'un réseau de neurones récurrent (RNN) utilisant des couches de type GRU (Gated recurrent units) et un mécanisme d'attention combiné à des techniques d'échantillonnage sur la distribution produite après l'entraînement, ainsi que des techniques d'augmentation de données.

3.1 Données d'entraînement

Les données d'entraînement ont joué un rôle crucial dans notre étude, fournissant les fondements nécessaires à l'apprentissage de nos modèles de génération de mélodies à partir de paroles. Nous avons exploité un ensemble de données richement annoté, composé de séquences de notes MIDI jumelées avec des syllabes provenant de l'article sur les LSTM-GANs conditionels (Yu et al., 2021). Chaque séquence comprend des informations détaillées sur la note, sa durée et le silence précédent, permettant ainsi aux modèles d'apprendre les relations subtiles entre les paroles et les éléments musicaux. Cette approche nous a permis de capturer la complexité des interactions entre les aspects textuels et musicaux d'une composition, offrant ainsi une base solide pour la génération de mélodies cohérentes et expressives. En exploitant ces données riches et diversifiées, nous avons pu entraîner notre modèle à générer des mélodies qui sont non seulement fidèles aux paroles fournies, mais également agréables à entendre.

3.2 Random Baseline

Dans le cadre de notre configuration expérimentale, nous avons mis en place un générateur aléatoire afin d'établir une référence pour la performance de génération de mélodies. Ce générateur de référence fonctionne en échantillonnant de manière aléatoire des triplets MIDI (note, durée, pause) à partir des distributions de probabilité observées dans l'ensemble de données d'entraînement issu de l'article sur les LSTM-GANs. En reproduisant les propriétés statistiques des données d'entraînement, ce modèle de référence génère des mélodies de manière purement stochastique, sans aucune compréhension contextuelle ou cohérence sémantique. Bien que simple, ce générateur de référence aléatoire fournit un point de comparaison essentiel pour évaluer la performance des modèles de génération de mélodies plus avancés.

3.3 T5 Transformer

Nous avons initialement exploré l'architecture Transformer T5 de Google (Raffel et al., 2023), dotée d'un encodeur pré-entraîné adapté à nos spécificités via un modèle encodeur-décodeur. L'approche initiale, utilisant des tokens spéciaux (`<int>note_<float>dur_<float>gap`) pour les entrées et sorties, s'avérant inefficace, nous avons opté pour un décodeur personnalisé à trois couches de sortie pour les notes, les durées et les pauses, permettant une manipulation fine des hyperparamètres et du vocabulaire de sortie (1b). Nos prédictions, basées sur des vocabulaires numériques pour les notes (0 à 128), durées (0 à 18) et pauses (0 à 7), ont d'abord conduit à un sur-apprentissage et des résultats monotones. Pour remédier à cela, nous avons intégré des techniques de codage positionnel et de masquage dans notre modèle, afin de prévenir la fuite d'informations futures pendant les prédictions:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{modèle}}}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{modèle}}}}\right)$$

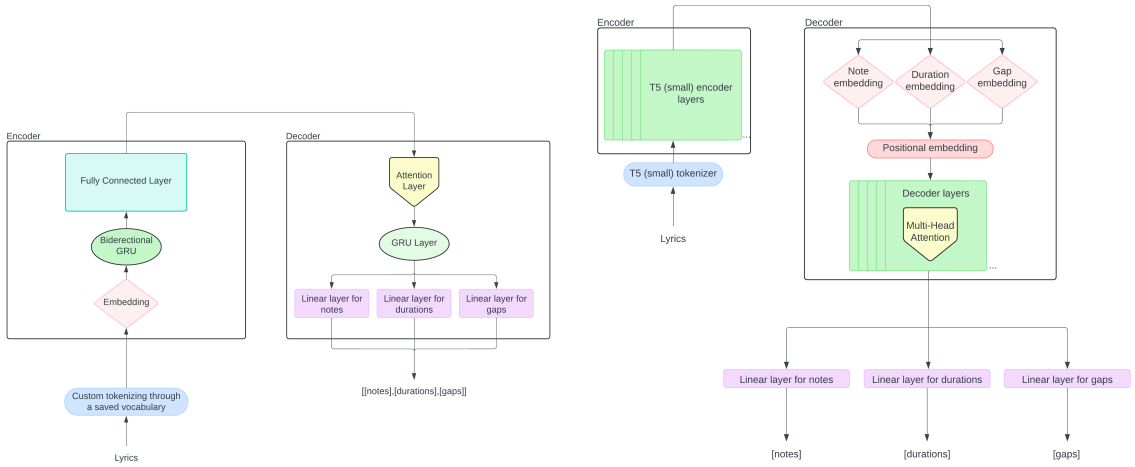
où *pos* est la position dans la séquence, i est l'indice de dimension, et $d_{\text{modèle}}$ est la taille de l'incorporation. Ces incorporations sont ensuite ajoutées aux incorporations d'entrée pour fournir des informations de position relative.
Masquage:

- Une matrice triangulaire inférieure est utilisée pour le masque cible. Cette matrice est de taille trg_len × trg_len, où 'trg_len' est la longueur de la séquence cible.
- La partie triangulaire inférieure (y compris la diagonale) est remplie de uns, permettant au modèle de se concentrer sur les positions précédentes et la position actuelle, mais pas sur les positions futures.

Le modèle utilise l'optimiseur AdamW avec des hyperparamètres spécifiques: une taille de lot de 1024, 200 époques, un taux d'apprentissage de 2e-7, et un poids de décomposition de 0.04. Nos couches d'encodeur et de décodeur possèdent chacune une taille cachée de 64, une dimension d'embedding de 64, quatre couches, un facteur d'expansion de 2 et huit têtes d'attention, avec un taux de dropout élevé à 0.6 pour réguler le sur-apprentissage. Les poids des pertes pour les notes, durées et pauses sont respectivement de 1.25, 0.5, et 0.3. Cependant, malgré ces ajustements, beaucoup d'optimisation de hyperparamètres et la simplification du décodeur, la complexité des résultats reste élevée, probablement due à la taille relativement petite de notre ensemble d'entraînement comparé aux standards habituels des modèles transformeurs.

3.4 RNN avec couches GRU

Pour résoudre les problèmes de surapprentissage rencontrés, nous avons exploré l'utilisation d'un modèle moins complèxe basé sur des couches GRU (Gated Recurrent Unit) (Cho et al., 2014) avec un mécanisme d'attention basé sur Bahdanau et al. (2016) pour le décodeur. Notre modèle consiste en un encodeur bidirectionnel avec une couche GRU de 400 unités, suivi d'un décodeur unidirectionnel de même taille. Nous avons opté pour un apprentissage de plongements (128 dimensions) plutôt que d'utiliser Word2Vec (Yu et al., 2021), ce qui nous permet d'obtenir des plongements plus pertinents et spécifiques au modèle pendant l'entraînement. De plus, nous avons intégré un mécanisme d'attention pour sélectionner les syllabes les plus pertinentes afin de générer chaque note MIDI. Pour régulariser le modèle, un dropout de 0.5 a été appliqué. Le taux d'apprentissage a été fixé à 0.00001 sur 200 epochs, avec un arrêt anticipé autour de 30 epochs. Nous avons utilisé un batch size de 1024 et optimisé le modèle avec l'algorithme AdamW avec un weight decay de 0.01. (1a)



(a) RNN Architecture (b) Transformer Architecture

Figure 1: Comparaison des architectures

3.5 Augmentation des données

Pour améliorer la diversité et la robustesse de notre modèle de génération de mélodies, nous avons mis en œuvre des techniques d'augmentation de données. Ces techniques visent à enrichir l'ensemble d'entraînement en introduisant des variations artificielles dans les données existantes, ce qui peut aider à prévenir le sur-apprentissage et à améliorer les performances du modèle lors de la généralisation à de nouvelles données.

1. **Fênetre glissante:** Une technique d'augmentation de données couramment utilisée est l'utilisation d'une fenêtre glissante. Dans notre approche, nous avons implémenté une fenêtre glissante avec un pas de taille 20. Cette technique consiste à diviser la séquence de données en segments de longueur fixe (20 dans notre cas) en faisant glisser une fenêtre le long de la séquence avec un pas fixe. En appliquant cette technique, nous avons pu augmenter la taille de l'ensemble d'entraînement tout en préservant la structure séquentielle des données musicales. Le choix d'un pas de taille 20 a été délibéré pour éviter de créer un ensemble d'entraînement trop volumineux, tout en introduisant suffisamment

de variations pour enrichir l'apprentissage du modèle. Avec cette technique, la taille des données d'entraînement est passé de 11 149 à 58 579 paires de paroles-notes.

2. **Transposition de notes:** Une autre technique d'augmentation de données que nous avons employée est la transposition des notes. Cette technique consiste à modifier la hauteur des notes dans les mélodies en décalant leur valeur de quelques tons vers le haut ou vers le bas. Les transformations sont appliquées de manière aléatoire sur un pourcentage de chacun des lots lors de l'entraînement, afin de créer des variations différentes à chaque itération. En appliquant des transpositions aléatoires aux mélodies de l'ensemble d'entraînement, nous avons introduit une variabilité supplémentaire dans les données, ce qui peut aider le modèle à apprendre des motifs musicaux plus robustes et à généraliser efficacement à des séquences de notes de différentes hauteurs.

3.6 Génération

Nous avions initialement utilisé l'opération argmax pour sélectionner la note la plus probable à chaque pas de temps. L'utilisation exclusive de l'opération argmax pour sélectionner la note la plus probable à chaque étape de la génération a conduit à des résultats répétitifs, où les mêmes valeurs étaient souvent générées de manière récurrente. Cette limitation a motivé l'adoption de techniques d'échantillonnage, de sélection des meilleurs candidats (topk) et de contrôle de la température pour diversifier les prédictions et introduire davantage de variabilité dans les séquences générées. Voir Appendix 4 et Appendix B

3.7 Évaluation

Dans la section d'évaluation, nous avons utilisé diverses méthodes pour évaluer les performances de nos modèles par rapport à la ligne de base aléatoire et aux LSTM-GANs. Ces techniques d'évaluation comprenaient des scores BLEU-n modifiés pour évaluer la similarité des transitions de notes, des durées et des intervalles entre les mélodies générées et les données de référence. Afin de standardiser les métriques et de produire des résultats comparables aux résultats de Yu et al. (2021) l'analyse s'est restreinte à des séquences de 20 triplets (note, longueur, silence)

3.7.1 Score BLEU. Le score BLEU est traditionnellement utilisé dans l'évaluation de traductions de textes entre les langues. Dans l'optique où notre modèle est tâché de «traduire» de la *langue* des paroles vers la *langue* des mélodies, il est approprié de considérer sa valeur; du moins à fins de comparaison avec la plupart des autres publications.

Le BLEU-n a été calculé avec la formule:

$$\text{BLEU-n} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log(p_n)\right)$$

où $\text{BP} = \exp\left(\min(1 - \frac{\text{refsLen}}{\text{candidateLen}}, 0)\right)$, répresente une pénalité pour les transitions courts et $p_n = \frac{\text{ClippedCounts}_n}{\text{Total Counts}_n}$ répresente la précision des n-grammes, qui est le rapport entre le nombre de n-grammes correspondants dans la traduction générée et le nombre total de n-grammes dans la traduction générée.

Dans notre cas, l'étude des scores BLEU a été effectuée sur les composantes mélodiques individuelles, soit pour le rythme, pour les silences et pour les transitions de notes. L'approche légèrement moins standard de considérer les transitions de notes plutôt que les notes elles

mêmes vient du fait que notre modèle ne génère pas des mélodies dans une tonalité commune aux mélodies de référence; une ceraine intuition pour le choix peut être gagnée en considérant que l'action de transposition d'une mélodie produirait un score BLEU très faible alors que la transposition est une classe d'équivalence entre les mélodies.

3.7.2 Discreptance maximale moyenne (MMD). De plus, nous avons utilisé la Discrepance Maximale Moyenne (MMD) comme métrique pour la sélection des meilleurs modèles lors de l'entraînement, ce qui nous a permis d'identifier les itérations de modèles les plus performantes. Voici sa formule:

$$\mathrm{MMD}(X, Y) = u_{XX} - u_{YY} - 2u_{XY}$$

Où u_{XY} réprésente la moyenne de noyau entre X et Y. Le noyau utilisé a été un Gaussian (RBF), où $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$.

En effet, cette métrique nous permet d'évaluer la similitude entre deux distributions. Il est attendu que deux mélodies associées à un même ensemble de paroles partagent la même distribution.

3.7.3 Métriques de répétitions. Puisque la musique est de nature répétitive, soit par le rythme ou par les techniques de composition usuelles, il est important de considérer comment les mélodies étudiées se comportent sous différents critères de répétition.

- **n-notes MIDI répétés:** le nombre d'apparition de chaque n-gramme dans une mélodie. Dans une mélodie, on s'attend à ce que des sections soient réutilisées, d'où l'importance de considérer cette métrique.
- **nombre de transitions des notes MIDI:** le nombre de fois que chaque saut entre deux notes effectué. Il n'est question que de la valeur de la différence entre deux notes succéssives et donc se distinc de la simple analyse de bi-grammes. L'oreille musicale porte beaucoup attention aux composantes relatives au contexte musical présent; en soi, sans être une répétition de bi-gramme, un enchainement de transitions déjà entendu apporte le même genre de qualité.

3.7.4 Métriques de distribution. La plupart des métriques soulignées sont pour la plupart trop faibles pour être indicatives de quoi que ce soit; en ce sens, elles sont quasi uniquement utiles à être comparées ensemble aux distributions organiques.

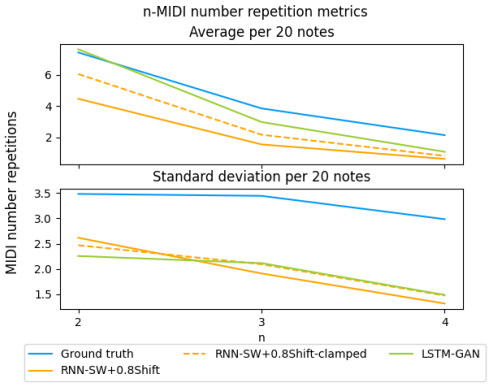
- **Étendu de notes MIDI:** correspond à différence entre la plus haute note et la plus basse note dans une mélodie. Cette métrique permet d'évaluer la capacité du modèle à créer des mélodies se conforment aux habitudes de tessiture humaine.
- **Nombre de notes MIDI différentes:** simplement le nombre de valeurs MIDI uniques dans une mélodie. Sert pricipalement à être comparée avec une distribution organique, mais peut grossièrement être un indicateur de qualité: 20 notes différentes sur 20 notes n'est probablement pas musical.
- **Nombre de notes non suivies d'un silence:** simplement le nombre de notes qui ne sont pas suivies d'un silence.
- **Valeur de silence moyenne:** moyenne des valeurs de silence au sein d'une mélodie.
- **Longueur de la mélodie:** somme des valeurs de longueur de chaque note avec la somme des silence les séparant.

- **Distribution des valeurs de notes MIDI:** la distribution des occurences de chaque note MIDI dans une mélodie.
- **Différence avec la gamme la plus proche:** parmis les 12 gammes majeures, mineures et mineures harmoniques, on calcule la somme des différences entre les notes présentes et les notes de la gammes, podérés par la longueur de chaque note. Une valeur trop haute signifie clairement que le modèle n'a pas appris le concept de tonnalité, ce qui réduit énormément les chances que la mélodie soit agréable.

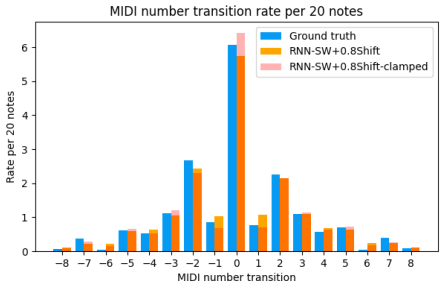
4 Résultats

4.1 Rammener à la plus proche tonalité

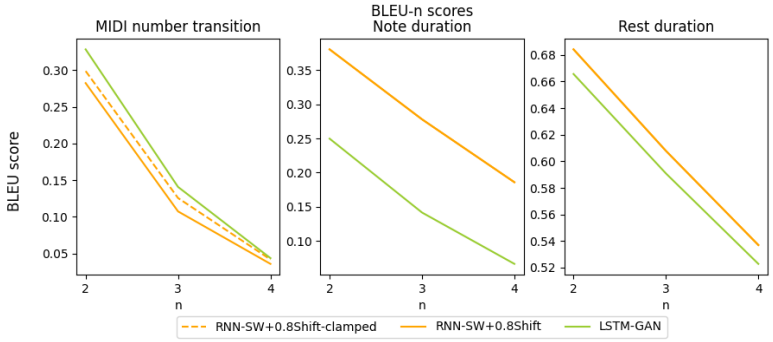
Dans la publication de Yu et al. (2021) il est question de rammener les mélodies générés à la tonalité qui lui est le moins destructive, soit la tonnalité minimisant la différence de gamme. Nos résultats montrent que rammener à la plus proche gamme a un effet positif sur la qualité des mélodies générés. Vois Figurer 2 pour l'analyse. On constate notamment



(a) Comparaison du nombre de n-notes MIDI répétés



(b) Comparaison de la distribution des transitions de notes MIDI



(c) Comparaison des scores BLEU-n

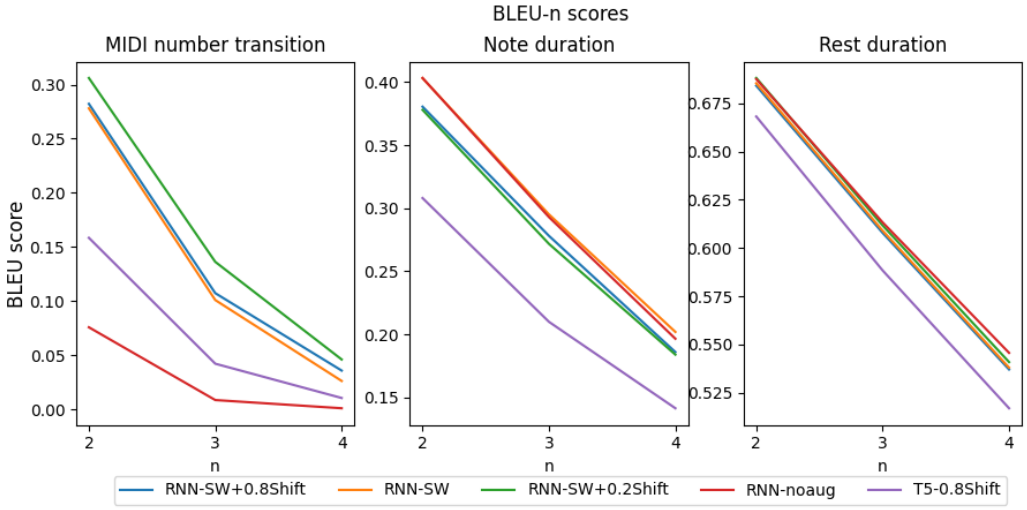
Figure 2: Effet de rammener à la gamme sur les métriques de répétitions

une augmentation du score BLEU et un rapprochement des métriques de répétitions à celles trouvées dans la distribution organique. Ces deux augmentations peuvent être attribuées au fait qu'en rammenant à la gamme, on réduit le nombre de notes uniques; soit que probabilistiquement, il y a plus de chances de retrouver un n-gramme ou une transition plus

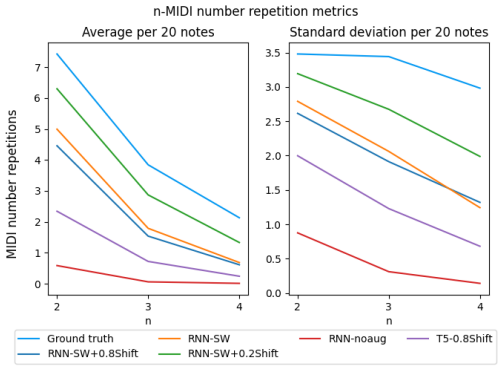
souvent. De plus on peut sensiblement constater que la distribution des transitions générée par notre modèle semble plus proche de la distribution de notre ensemble de test.

4.2 Augmentation des données

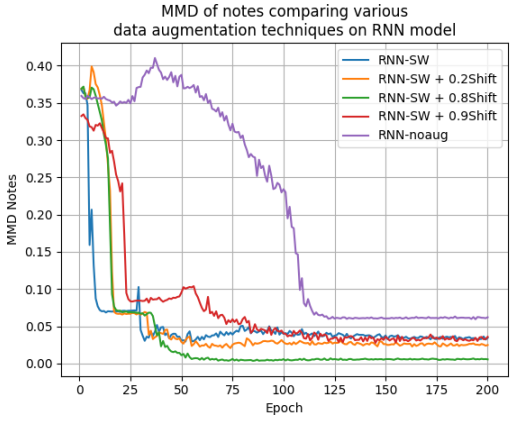
L'augmentation de données a été un succès pour notre travail, puisqu'elle aura permis l'obtention de meilleures scores BLEU ainsi que des valeurs de répétitions de notes MIDI plus proches des valeurs obtenues dans l'ensemble de test. On remarque aussi la particularité que l'ordre de classement des scores bleus pour les notes et les longueurs de notes sont grosièrement inversées. Nos modèles ont donc tendance à échanger de la capacité sur les rythmes pour de la capacité sur les notes. De plus, l'augmentation de données aura aussi permis d'obtenir des courbes de MMD sur l'entrainement ayant un plateau plus bas et donc de meilleure qualité. Voir la Figure 3.



(a) Comparaison des scores BLEU-n



(b) Comparaison du nombre de n-notes MIDI répétés



(c) Comparaison des courbes de MMD à l'entrainement

Figure 3: Effet des stratégies d'augmentation des données

4.3 Meilleurs résultats

Au final notre meilleur résultat aura été obtenu avec le modèle RNN et un augmentation des données avec la fenêtre coulissante et 20% de mélodies transposées; qui est ensuite rammené à la tonalité la plus proche. En effet celui-ci réussit a battre les résultats de LSTM-GAN sur les métriques concernant les notes, en particulier pour les répétitions plus longues et donc plus complèxes, mais avec plus de difficulté pour les métriques de temps comme la longueur des notes et des silences. Voir Table 1. Pour écouter les musiques générés voir Appendix B.

Metric	**Ground Truth**	**LSTM-GAN**	**RNN-SW** 0.2Shift+clamped	**Random**
2-MIDI num. reps.	7.424	7.628	7.222	0.208
3-MIDI num. reps.	3.843	2.971	3.262	0.004
4-MIDI num. reps.	2.131	1.067	1.516	0
MIDI num.s span	10.776	7.671	10.354	39.206
Unique MIDI nums.	5.884	5.09	5.632	14.744
Notes no rest	15.985	16.708	19.308	15.626
Avg. rest value in song	0.776	0.034	0.076	0.787
Song length	43.254	30.108	22.32	14.157
Scale diff.	0.01	0.0	0.0	0.169
BLEU-2 Notes	NA	0.32	0.31	0.03
BLEU-3 Notes	NA	0.14	0.14	0
BLEU-4 Notes	NA	0.04	0.05	0

Table 1: Comparaisons des métriques pour notre meilleur modèle

5 Conclusions

Notre projet contribue à améliorer l'expression artistique et l'innovation dans la création musicale. En automatisant le processus de génération de mélodies, il permet aux artistes et musiciens de se concentrer sur les aspects créatifs de leur travail, tels que les paroles et les arrangements, rationalisant ainsi le processus de composition de chansons et potentiellement réduisant les barrières pour les nouveaux venus. Cette démocratisation de la production musicale peut conduire à une gamme plus diversifiée de voix et de styles dans l'industrie, enrichissant le paysage culturel.

De plus, en fournissant des outils qui aident à générer des compositions musicales, notre projet peut également servir à des fins éducatives, aidant les étudiants et les musiciens en herbe à comprendre la structure musicale et les techniques de composition. Cela peut inspirer et encourager davantage de personnes à s'engager dans la création musicale, favorisant ainsi la créativité et l'appréciation culturelle à une échelle plus large.

Finalement, notre meilleur modèle présente l'avantage significatif de ne pas être aussi lourd que celui du papier LSTM-GAN tout en offrant des performances similaires en termes de création, de respect de la structure et du rythme musicaux. Cela se traduit par une plus grande efficacité et accessibilité, permettant une utilisation plus large, même avec des ressources informatiques limitées. Ce progrès technologique souligne notre engagement envers l'innovation durable et inclusive dans le domaine de la création musicale.

6 Contributions de chaque membre de l'équipe

- **Jaydan Aladro:** Implémentation du modèle Transformeur décodeur bâti par-dessus l'encodeur de T5 et rafinement des hyperparamètres s'y rattachant (et les autres tentatives d'utilisation des Transformeur). Script de l'entraînement des modèles. Script de la génération des modèles. Collateur des données.
- **Maxime Bélanger:** Évaluation et tests approfondis de nos modèles et des modèles de références afin de comparer les résultats et fournir une discussion détaillée. Génération des visualisations de données pour le rapport et pour les choix de dévellopement. Source de connaissance musicales. Implémentation des project_utils. Choix des métriques d'évaluation.
- **Guillaume Charron:** Implémentation du modèle RNN encodeur décodeur personnalisé et rafinement des hyperparamètres s'y rattachant. Ajout des méthodes d'échantillonage, de topK et de température à la génération des prédictions pour améliorer les résultats. Ajout de l'augmentation de données basé sur le décalage aléatoire de tons. Collateur des données.
- **Jeremy Kaufman:** Implémentation de l'augmentation de données basée sur une fenêtre glissante pour réduire le sur-apprentissage et du modèle aléatoire de référence afin de pouvoir comparer nos modèles avec une référence aléatoire. Exploration des données et mis en place du loading du data. Majorité de la rédaction du rapport.

Appendix

A Additional Visualisation

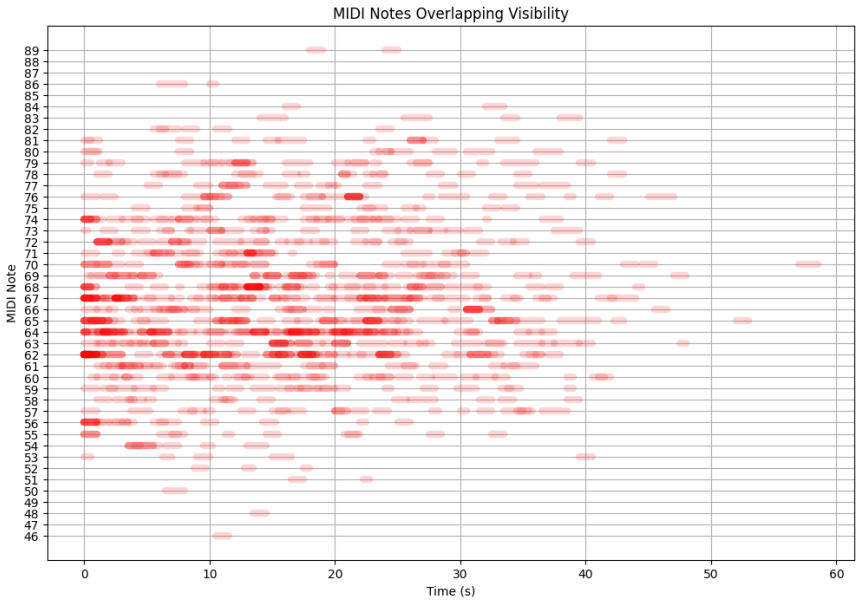


Figure 4: Overlapping of 20 different inferences from our best model on the same lyrics. Temperature = 1. Top-30.

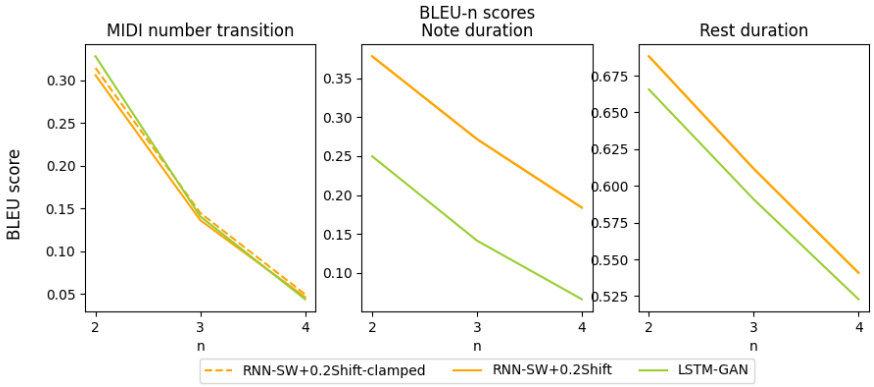


Figure 5: Scores BLEU pour le meilleur modèle

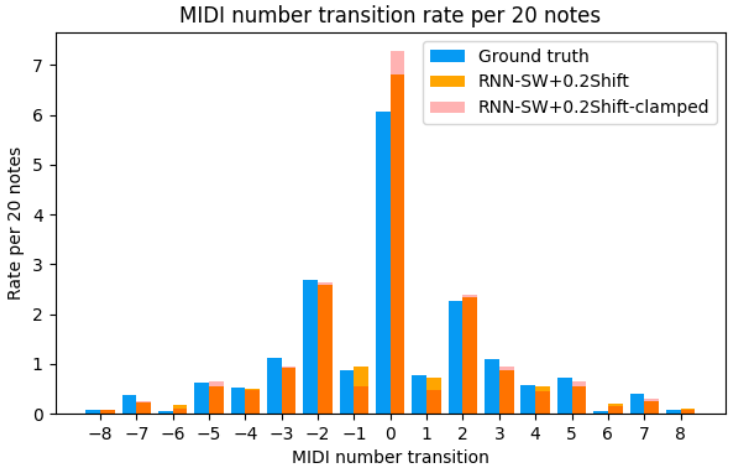


Figure 6: Distribution des transitions pour le meilleur modèle

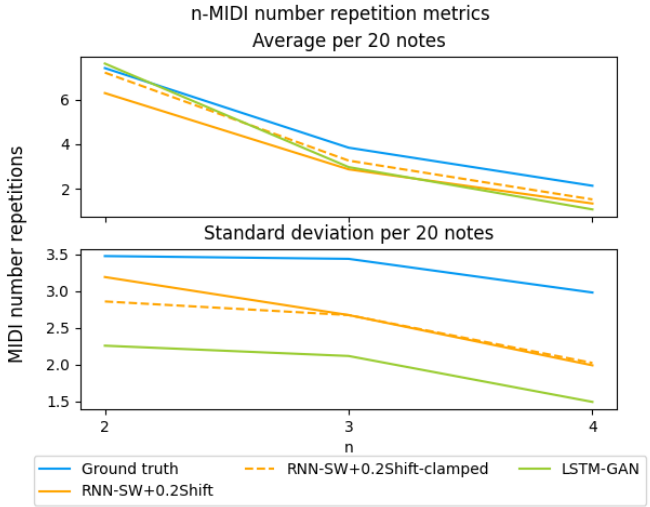


Figure 7: Métriques de répétitions pour le meilleur modèle

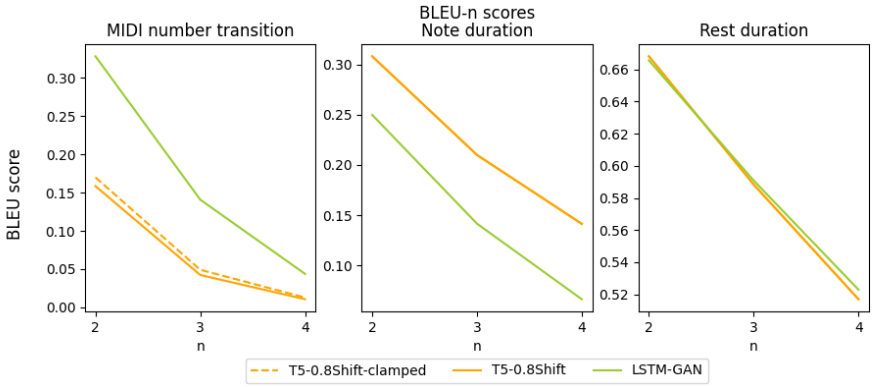


Figure 8: Scores BLEU pour le modèle basé sur T5

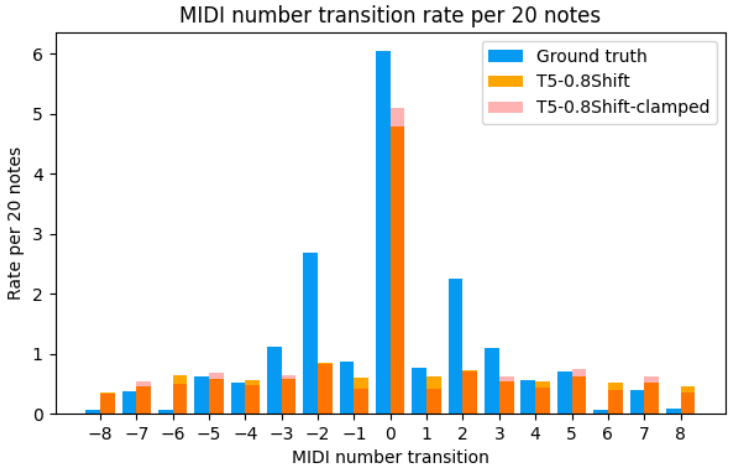


Figure 9: Distribution des transitions pour le modèle basé sur T5

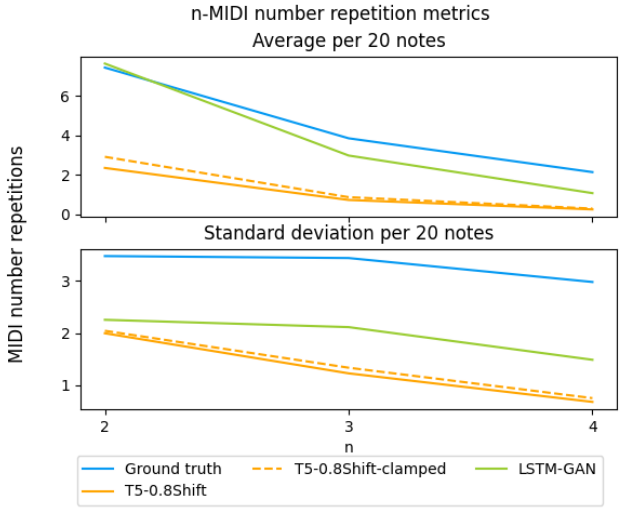


Figure 10: Métriques de répétitions pour le modèle basé sur T5

B Generated Musics

RNN avec 20% shifting and meilleur transformeur: Cliquez ici

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2016.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

M, G. R., Zhang, Z., Yu, Y., Harscoet, F., Canales, S., and Tang, S. Deep attention-based alignment network for melody generation from incomplete lyrics, 2023.

Nichols, E., Morris, D., Basu, S., and Raphael, C. Relationships between lyrics and melody in popular music., 01 2009.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

Yu, Y., Srivastava, A., and Canales, S. Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 17(1):1–20, February 2021. ISSN 1551-6865. doi: 10.1145/3424116. URL `http://dx.doi.org/10.1145/3424116`.