République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ SAAD DAHLEB – BLIDA 1

Faculté de science

Département d'informatique

Mémoire de Projet de Fin d'Études

Pour l'obtention du diplôme de Master en Informatique

Option : Ingénierie du Logiciel

TITRE

Moteur de recherche audio basé sur des requêtes textuelles

Présenté par

Yahiaoui Yahia

Proposé par

Kameche Abdallah Hicham

Soutenu le : 21 Juin 2023.

Devant le jury composé de

Benaissi Sellami, Université de Blida 1, Blida.	**Président**
Boucetta Zouhel, Université de Blida 1, Blida.	**Examinatrice**
Kameche Abdallah Hicham, Université de Blida 1, Blida.	**Promoteur**

Année Universitaire 2022-2023

RÉSUMÉ

La récupération audio basée sur la langue est un type de système de récupération d'informations qui permet aux utilisateurs de rechercher du contenu audio à l'aide de requêtes en langage naturel. Cette tâche a reçu beaucoup d'attention ces dernières années, car cette technologie a de nombreuses applications, notamment dans les domaines du divertissement, de l'éducation et de la santé. Pour réaliser notre vision, nous avons effectué plusieurs tests en utilisant une méthode alternative pour valider nos résultats, nous avons utilisé l'ensemble de données de légende phonétique et converti ses phrases en vecteurs grâce à l'utilisation de sBert/Bert. De plus, nous avons extrait le spectrogramme log mel des fichiers audio correspondants. Pour approfondir davantage notre analyse, nous avons appliqué différentes architectures de réseau de neurones convolutifs (CNN) et de réseau de neurones récurrents (RNN) pour extraire des caractéristiques du diagramme de spectre log mel ou pour générer une prédiction spectrogramme log mel, nous avons calculé leur similarité avec le sous-titre à l'aide de points similarité du produit ou du cosinus, puis nous avons appliqué certaines matrices pour l'évaluation et comparé le résultat avec d'autres résultats

Mots-clés: Récupération audio basée sur la langue, requêtes en langage naturel, spectrogramme log mel, Bert, sBert

ABSTRACT

Language-based audio retrieval is a type of information retrieval system that allows users to search audio content using natural language queries. This task has received a lot of attention in recent years, as this technology has many applications, especially in the fields of entertainment, education and health. To achieve our vision, we ran several tests using an alternative method to validate our results, we used the phonetic caption dataset and converted its sentences into vectors through the use of sBert/Bert. In addition, we extracted the log-mel-spectrogram from the corresponding audio files. To deepen our analysis further, we applied different architectures of convolutional neural network (CNN) and recurrent neural network (RNN) to extract features from log-mel-spectrogram or to generate a log-mel-spectrogram prediction, we calculated their similarity to the caption using dot product or cosine similarity, then We applied some matrices for evaluation and compared the result with other results

Keywords: Language-based audio retrieval, Natural language queries, log mel spectrogram, Bert, sBert

ملخص

استرجاع الصوت المستند إلى اللغة هو نوع من نظام استرجاع المعلومات الذي يسمح للمستخدمين بالبحث في المحتوى الصوتي باستخدام استعلامات اللغة الطبيعية. حظيت هذه المهمة باهتمام كبير في السنوات الأخيرة ، حيث أن لهذه التقنية تطبيقات عديدة ، خاصة في مجالات الترفيه والتعليم والصحة. لتحقيق رؤيتنا ، أجرينا العديد من الاختبارات باستخدام طريقة بديلة للتحقق من صحة نتائجنا ، واستخدمنا مجموعة بيانات التسمية التوضيحية الصوتية وقمنا بتحويل جملها إلى متجهات من خلال استخدام sBert / Bert. بالإضافة إلى ذلك ، قمنا باستخراج المخطط الطيفي log-mel من ملفات الصوت المقابلة. لتعميق تحليلنا بشكل أكبر ، قمنا بتطبيق بنيات مختلفة للشبكة العصبية التلافيفية (CNN) والشبكة العصبية المتكررة (RNN) لاستخراج ميزات من مخطط طيف log-mel أو لإنشاء تنبؤ مخطط طيفي لوغاريتمي ، و قمنا بحساب تشابهها مع التسمية التوضيحية باستخدام المنتج النقطي أو تشابه جيب التمام ، ثم طبقنا بعض المصفوفات للتقييم ومقارنة النتيجة بالنتائج الأخرى

الكلمات الرئيسية: استرجاع الصوت على أساس اللغة، استعلامات اللغة الطبيعية، sBert، Bert ،المخطط الطيفي log-mel، الشبكة العصبية التلافيفية (CNN)، الشبكة العصبية المتكررة (RNN)

Reconnaissance

Nous commençons par exprimer nos sincères remerciements à Dieu, car sans ses bénédictions et ses conseils, ce travail n'aurait pas été possible.

Nous tenons également à exprimer notre gratitude à notre encadrant Professeur ABDALLAH HICHAM KAMECHE de l'Université de Blida, pour ses précieux conseils et son soutien tout au long de nos recherches. Malgré nos connaissances limitées dans le domaine du traitement audio, il était toujours disponible pour nous fournir des informations et des commentaires précieux, et son souci du détail a été crucial pour mener à bien ce travail.

Nous remercions également les membres du jury pour le temps et les efforts qu'ils ont consacrés à l'examen de notre travail et à la fourniture de précieuses suggestions.

Enfin, nous tenons à remercier toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail de recherche. Nous reconnaissons et apprécions votre contribution.

Contenu

Liste des abréviations

Abréviation	Description complète
ADC	Analogue-to-digital-Converter
AE	Amplitude Enveloppe
AI / IA	Artificiel intelligence
BERT	Bidirectional Encoder Representations from Transformers
CNN [18]	Convolutional neural network
DFT	discrete fourier transform
DL	deep learning
EDA	Easy Data Augmentation
FFT	Fast fourier transform
FN	False negative
FP	False positive
FT	Fourier transform
IFT	Inverse Fourier transform
MAP	mean average precision
MFCC	Mel-frequency cepstral coefficients
ML	machine learning
NLP / PNL / TAL	natural language processing
RMS	Root-mean-square
RNN [17]	Recurrent neural network
sBERT	Sentence-BERT
STFT	short-time Fourier transform
TN	True negative
TP	true positive
ZCR	The Zero-Crossing Rate

Liste des figures

Liste des Tables

Introduction

1. Motivations

La possibilité de générer des représentations audio du texte peut transformer la façon dont nous percevons le monde qui nous entoure à travers le son. Cette technologie nous permet de convertir le texte écrit en sons qui imitent divers objets et éléments du monde, comme le son d'un oiseau qui chante, le moteur d'une voiture qui tourne ou le bruit de la pluie qui tombe. Cette technologie révolutionnaire a le potentiel d'améliorer les expériences de réalité virtuelle, de jeux vidéo et d'autres formes de médias en offrant un environnement audio plus immersif et réaliste. De plus, il peut être utilisé pour créer des descriptions audio de contenu visuel pour les personnes ayant une déficience visuelle, leur permettant de découvrir le monde à travers le son. Le développement de cette technologie nécessite la création d'une vaste base de données de sons pouvant être associés à différents mots et phrases, permettant au système de générer avec précision le son approprié. Les chercheurs et les développeurs travaillent activement à faire progresser cette technologie et à la rendre largement disponible pour diverses applications dans des domaines tels que le divertissement, l'éducation et la santé. Cela peut ouvrir de nouvelles opportunités d'interaction homme-machine et révolutionner la façon dont nous percevons et interagissons avec le monde à travers le son.

2. Contributions

La technologie text-a-audio, qui génère des représentations sonores du texte, a le potentiel d'apporter des contributions significatives dans divers domaines, notamment la recherche, le divertissement, l'éducation et l'accessibilité. En recherche, cette technologie peut être utilisée pour analyser et représenter des données textuelles de nouvelles façons. Par exemple, il peut être appliqué dans des domaines tels que la linguistique pour étudier les modèles sonores et la phonétique de différentes langues. Il peut également être utilisé en sciences sociales pour analyser et représenter de grandes quantités de données textuelles pour une analyse qualitative.

Dans le monde du divertissement, la technologie text-to-audio peut améliorer les expériences immersives dans les jeux vidéo, la réalité virtuelle et les environnements de réalité augmentée. Cette technologie peut fournir aux utilisateurs un environnement audio plus réaliste et engageant, leur permettant de se sentir plus connectés à l'expérience.

Dans le secteur de l'éducation, la technologie text-to-audio peut bénéficier aux apprenants en les aidant à développer leurs capacités d'écoute et leur prononciation. Il peut également être utilisé pour fournir des descriptions audio de contenus visuels dans des supports pédagogiques, les rendant ainsi plus accessibles aux personnes ayant une déficience visuelle.

De plus, la technologie text-to-audio peut jouer un rôle important en rendant le contenu plus accessible aux personnes handicapées. Cette technologie peut générer des descriptions audio de contenu visuel, permettant aux personnes ayant une déficience visuelle de découvrir le monde à travers le son. De plus, il peut être utilisé pour fournir un retour audio dans diverses applications, telles que les éditeurs de texte et les clients de messagerie, permettant aux personnes malvoyantes de naviguer et d'interagir avec le logiciel.

Dans l'ensemble, les contributions de la technologie text-to-audio sont importantes et de grande envergure, allant de la possibilité pour les chercheurs d'étudier le langage et les données sociales de nouvelles façons à la fourniture d'expériences immersives et accessibles dans divers domaines. Au fur et à mesure que la technologie progresse, elle a le potentiel de transformer la façon dont nous interagissons avec le monde qui nous entoure grâce au son.

3. Organisation de mémoire

Ce mémoire est divisé en trois chapitres. Le chapitre 1 donne un aperçu des principes fondamentaux du traitement du signal audio, du traitement du langage naturel et des travaux connexes. Le chapitre 2 décrit notre approche proposée, y compris les architectures utilisées. Le chapitre 3 présente les résultats de nos travaux, analyse les résultats et discute des travaux futurs.

Chapitre1: Traitement audio / Traitement de texte / travaux connexes

Introduction:

Dans ce chapitre, nous explorerons deux domaines importants du traitement numérique du signal: le traitement audio et le traitement de texte. Ces deux domaines impliquent l'utilisation d'algorithmes et de techniques pour manipuler et analyser les données, mais les types de données et d'applications sont différents.

Dans la première partie de ce chapitre, nous nous concentrerons sur le traitement audio. Nous commencerons par discuter des bases des ondes sonores, des signaux audio et des paramètres du son. Nous explorerons ensuite les différentes fonctionnalités des signaux audio et le pipeline d'extraction. Ensuite, nous nous plongerons dans les domaines temporel et fréquentiel des signaux audio, y compris la transformée de Fourier et ses différentes formes. Nous examinerons également les spectrogrammes Mel et les coefficients cepstraux de fréquence Mel (MFCC) en tant que caractéristiques importantes pour le traitement audio.

Dans la deuxième partie de ce chapitre, nous nous concentrerons sur le traitement de texte. Plus précisément, nous aborderons le domaine du traitement automatique du langage naturel (NLP) et ses différentes techniques d'analyse et de traitement de données textuelles. Nous explorerons les bases de la tokenisation, de la radicalisation et de la lemmatisation, ainsi que les techniques de traitement des mots vides et d'encodage de texte. Nous aborderons également le modèle de sac de mots (BOW), TF-IDF, les incorporations de mots et certains algorithmes populaires tels que Word2Vec [4], GloVe, BERT [20] et sBERT. [19]

Dans l'ensemble, ce chapitre fournira un aperçu des concepts et techniques fondamentaux utilisés à la fois dans le traitement audio et de texte, en mettant l'accent sur les applications pratiques et les exemples concrets.

Part 1: Traitement audio

1. Introduction:

Avec le développement et la diversité des moyens de communication, le son a joué un rôle important dans la compréhension et l'amélioration des données de ce processus, car il existait de nombreuses sources sonores, notamment les sons d'instruments de musique, la parole des gens, les sons de mouvement, les sons émis par divers choses, en plus des bruits qui rendent difficile la compréhension de ce qui se passe. D'où la nécessité de pratiquer le traitement de la voix

Dans ce chapitre, nous examinerons plusieurs manières dont ce traitement se produit, en commençant par le son et les formes d'onde, les paramètres des sons, la conversion analogique-numérique, puis nous examinerons certaines caractéristiques audio dans le domaine temporel et dans le domaine fréquentiel et nous terminerons par la visualisation du son. et la canalisation d'extraction

2. Son et formes d'onde :

Le son est généré par les vibrations d'un objet, provoquant l'oscillation et la collision des molécules d'air environnantes. Ces collisions, à leur tour, modifient la pression de l'air, donnant naissance à une vague. Les formes d'onde servent de moyen visuellement attrayant pour représenter et comprendre le son. En représentant graphiquement la variation de l'écart de pression par rapport au niveau zéro au fil du temps, les formes d'onde fournissent des informations précieuses sur la fréquence, l'intensité et la durée du son. Les ondes mécaniques, y compris les ondes sonores, se caractérisent par leur capacité à se propager dans l'espace, en transférant de l'énergie d'un point à un autre. Cependant, contrairement aux ondes électromagnétiques, les ondes mécaniques nécessitent un milieu, tel que l'air, dans lequel elles peuvent se déformer et se propager.

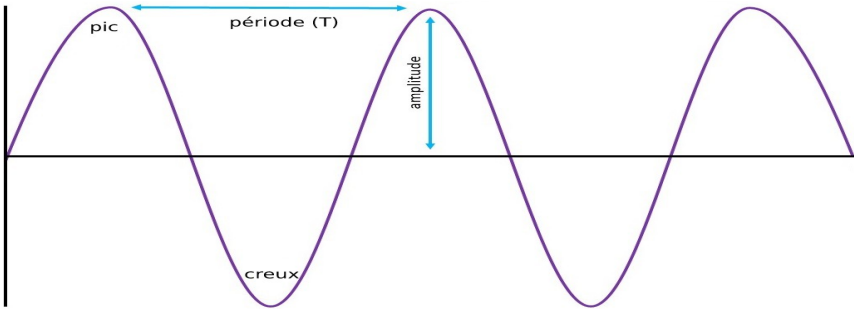
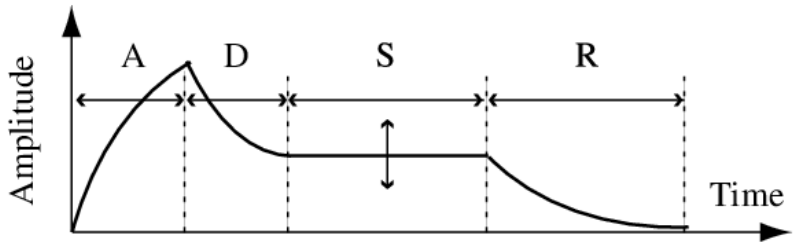


Figure 1: onde sonore

3. paramètres des sons:

paramètres	description
Période (T)	est la durée avant d'avoir 2 pics ou 2 creux
fréquence (f)	est l'inverse de la période et elle est exprimée en hertz (cycles par seconde), une fréquence plus élevée signifie un son plus élevé, 2 fréquences sont perçues de manière similaire si elles diffèrent d'une puissance de 2
la hauteur	est un concept logarithmique que nous utilisons pour la perception de la fréquence, l'octave est divisée en 1200 cents et la différence de hauteur perceptible est de 10-25 cents
L'amplitude (A)	est la hauteur ou la valeur de la perturbation de la pression atmosphérique et nous obtenons cette valeur en regardant de zéro au pic ou de zéro aux creux, une plus grande amplitude signifie un son plus fort
Phase (φ)	est une comparaison de la phase de deux formes d'onde, généralement de la même fréquence nominale. En temps et en fréquence, le but est de déterminer le décalage de fréquence (différence entre les cycles du signal)
puissance sonore	la puissance sonore en général est une mesure physique que nous pouvons exprimer comme la vitesse à laquelle l'énergie est transférée, pour être plus précis c'est l'énergie par unité de temps émise par une source sonore dans toutes les directions et elle est mesurée en watt (W)
intensité sonore	l'intensité sonore est la puissance sonore par unité de surface mesurée en W/m^2 , Le niveau d'intensité est une échelle logarithmique mesurée en décibels (dB) et c'est le rapport entre deux valeurs d'intensité, Chaque ~3 dBs double l'intensité, nous décrivons les décibels avec: $dB(I) = 10. \log_{10}(I/I_{TOH})$
Seuil	L'être humain peut percevoir des sons avec de très petites intensités

d'audition	TOH = 10^{-12} W/m^2
Seuil de douleur	intensités sonores qui causent de la douleur à l'homme TOP = 10.W/m^2
Loudness	est une perception subjective de l'intensité sonore qui dépend de la durée/fréquence d'un son et de l'âge des personnes, mesurée en phons
Timbre	il n'y a pas de définition réelle et complète du timbre, mais nous pouvons nous y référer comme la couleur du son, le timbre est la différence entre deux sons avec la même intensité, fréquence, durée. Décrit avec des mots comme : brillant, sombre, terne, dur, chaleureux, Timbre est multidimensionnel caractérisé par: Enveloppe sonore, Contenu harmonique, Modulation d'amplitude/fréquence
Enveloppe sonore	le son a généralement une enveloppe qui peut être divisée avec un modèle appelé modèle ADSR qui signifie Attack-Decay-Sustain-Release Model -Attaque: est le pic initial d'amplitude sonore -Decay: c'est là que le son se stabilise -Sustain: c'est là où le son reste constant en amplitude -Release: c'est la phase de fondu du son  **Figure 2: Attack-Decay-Sustain-Release Model**
Contenu harmonique :(overtones)	le son complexe : est fait de superposition de sinusoïdes ces différentes sinusoïdes sont superposées pour créer un son complexe appelé « partiel », le partiel le plus bas est appelé fréquence fondamentale Un partiel harmonique est une fréquence qui est un multiple de la fréquence fondamentale, tous les sons ne sont pas parfaitement harmoniques, certains sont inharmoniques, ce qui indique une déviation par rapport à un partiel harmonique
La modulation d'amplitude	connu sous le nom de "tremolo", c'est une variation périodique de l'amplitude, en musique il est utilisé à des fins expressives
Modulation de fréquence	connu sous le nom de “vibrato”, c'est une variation périodique de fréquence, en musique elle est utilisée à des fins expressives
Signal audio	est une représentation possible d'un son il contient toutes les informations dont nous avons besoin pour reproduire le son une fois de plus nous pouvons diviser en 2 catégories
signal analogique	a des valeurs continues de temps sur l'axe des x et des valeurs continues d'amplitude sur l'axe des y, cette résolution infinie nécessite une mémoire infinie pour la stocker,

	nous devons donc passer au signal numérique
signal numérique	a une séquence de valeurs discrètes (point de données et) qui ne peut prendre qu'un nombre fini de valeurs, et pour cela vient l'ADC

Tableau 1 - paramètres des sons

4. conversion analogique-numérique (ADC): on l'appelle aussi avec modulation par impulsions codées, composée de 2 étages:

- échantillonnage : qui consiste à capturer des points de données d'une onde sonore à des intervalles spécifiques, déterminés par une période T choisie. Ces échantillons sont situés à des instants tn = n * T. Le taux d'échantillonnage, calculé comme sr = 1/T, détermine la nombre d'échantillons prélevés par unité de temps. Des taux d'échantillonnage plus élevés entraînent une réduction des erreurs. La fréquence de Nyquist, fn = sr/2, établit la plus grande fréquence qui peut être représentée avec précision. Les artefacts de repliement, qui sont des erreurs lorsque les fréquences supérieures sont représentées comme des fréquences inférieures, sont produits par des fréquences supérieures à la fréquence de Nyquist.

- la quantification: qui opère sur les valeurs d'amplitude en ordonnée plutôt que sur les valeurs de temps en abscisse, ce qui est différent de l'échantillonnage. L'axe y a un nombre fixe de valeurs d'amplitude discrètes, et à chaque échantillon, la valeur d'amplitude est approximée à la valeur disponible la plus proche sur l'axe y. La résolution de la quantification est généralement mesurée en nombre de bits utilisés.

5.fonctionnalités audio: les fonctionnalités audio sont une description du son où chacune nous fournit un aspect différent du son que nous pouvons utiliser pour entraîner nos systèmes audio intelligents, il existe quelques stratégies que nous pouvons utiliser pour catégoriser ces fonctionnalités :niveau d'abstraction, périmètre temporel, aspect musical, domaine du signal, approche ml. Nous pouvons classer les fonctionnalités (features) en deux categories: temporelles ou fréquentielles.

5.1 fonctionnalités du domaine temporel:

Le domaine temporel audio se réfère à l'analyse et au traitement des signaux sonores dans le temps. Il englobe un certain nombre de fonctionnalités et de techniques qui permettent de comprendre et de manipuler les caractéristiques temporelles d'un signal audio. Voici quelques fonctionnalités courantes du domaine temporel audio:

enveloppe d'amplitude (AE): c'est la valeur d'amplitude maximale de tous les échantillons d'une image, elle donne une idée approximative de l'intensité et de sa sensibilité aux valeurs aberrantes car nous pouvons avoir un pic qui n'est pas représentatif dans l'ensemble de l'image, nous l'utilisons principalement dans la détection d'apparition ou la musique classement des genres

Énergie quadratique moyenne (RMS): le nom dit tout, c'est une fonction qui prend l'énergie quadratique moyenne à une image t, cela indiquera le volume et c'est moins sensible aux valeurs aberrantes que AE parce que nous ne sommes pas en prenant une seule valeur d'échantillonnage à partir d'une image à la place, nous obtenons des informations de tout l'échantillon, il est principalement utilisé dans la segmentation audio, la classification des genres musicaux

Taux de passage par zéro (ZCR): il nous fournit le nombre de fois qu'un signal traverse l'axe horizontal, il est largement utilisé dans la reconnaissance vocale et le traitement de la musique, nous pouvons l'utiliser pour la reconnaissance des sons percutants par rapport à la hauteur ou dans l'estimation de la hauteur monophonique ou décision vocale/non prononcée pour les signaux vocaux

5.2. Fonctionnalités du domaine fréquentiel :

Le domaine fréquentiel en audio se réfère à l'analyse et au traitement des signaux sonores en termes de leur composition en différentes fréquences. Voici quelques fonctionnalités courantes du domaine fréquentiel :

Caractéristique	Description	Formule
la transformée de fourier	la FT permet de passer du domaine temporel au domaine fréquentiel en décomposant un signal sonore en ses composantes fréquentielles, représentées par des amplitudes et des phases associées à différentes fréquences. Cela permet d'analyser et de comprendre la composition fréquentielle d'un signal audio, la sortie d'une transformée de fourier est les coefficients de fourier qui fournissent des informations pour chaque fréquence où cela nous donne 2 paramètres l'un est la phase et l'autre est l'amplitude, l'amplitude nous indique la présence d'une certaine fréquence dans un signal d'origine	Équation des phases: $\varphi_f = argmax_{\varphi\in[0,1)}\left(\int s(t)\cdot sin(2\pi\cdot(ft-\varphi))\cdot dt\right)$ Équation de magnitude: $d_f = \max_{\varphi\in[0,1)}\left(\int s(t)\cdot sin(2\pi\cdot(ft-\varphi))\cdot dt\right)$ Formule FT (d): $\hat{g}(f) = \int g(t)\cdot e^{-i2\pi ft}dt$
transformée de fourier inverse (IFT):	nous pouvons reconstruire un signal en superposant toutes les sinusoïdes que nous avons extraites du signal d'origine, puis nous pondérons ces sinusoïdes par leur amplitude relative et en utilisant également la phase d'origine que nous avons extraite.	Formule IFT: $g(t) = \int c_f\cdot e^{i2\pi ft}df$
La transformée de fourier discrtète (DFT)	la DFT permet de passer d'un signal audio continu à un signal numérique discret en échantillonnant le signal d'origine. La DFT utilise une somme discrète pour calculer les composantes fréquentielles, en considérant un nombre fini d'échantillons. Dans la DFT, il existe une redondance due à une symétrie centrale. La moitié droite du spectre est le miroir de la moitié gauche. Pour nos besoins, nous nous concentrons généralement uniquement sur le côté	Formule DFT: $\hat{x}(k/N) = \sum_{n=0}^{N-1} x(n)\cdot e^{-i2\pi n\frac{k}{N}}$

	gauche de ce spectre, qui s'étend jusqu'à la fréquence de Nyquist (Sr/2), où Sr est le taux d'échantillonnage.	
transformée de fourier à court terme (STFT)	Le STFT permet d'analyser le contenu fréquentiel d'un signal dans le temps. C'est une technique qui fournit une représentation temps-fréquence d'un signal en calculant la transformée de Fourier sur de courts segments du signal qui se chevauchent.	formule de fenêtrage : $x_w(k) = x(k) \cdot w(k)$ Formule STFT: $S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$
Spectrogrammes mel	Le spectrogramme Mel nous permet d'analyser et de représenter le contenu fréquentiel d'un signal audio d'une manière plus conforme à la perception humaine du son. C'est une transformation du spectrogramme traditionnel qui utilise une échelle mel, qui est une échelle perceptuelle de hauteurs basée sur le système auditif humain	Fréquence hertz vers fréquence mel: $m = 2595 \cdot log(1 + \frac{f}{500})$ Fréquence mel vers fréquence hertz: $f = 700(10^{m/2595} - 1)$ Calcule de mel spectrogram: M = (# bands, framesize / 2 + 1) Y = (framesize / 2 + 1, # frames) Mel spectrogram = MY M: mel filter banks Y: spectrogram
Coefficients cepstraux Mel-fréquence (MFCC)	Les MFCC nous permettent d'extraire et de représenter des caractéristiques importantes d'un signal audio qui sont pertinentes pour la perception auditive humaine, pour l'appliquer, nous appliquons Mel-Scaling au cepstrum, puis nous appliquons la transformation discrète en cosinus	Formule de cepstrum $C(x(t)) = F^{-1}[log(F[x(t)])]$
le rapport d'énergie de bande (BER)	Le BER nous permet d'évaluer la répartition de l'énergie sur différentes bandes de fréquences dans un signal. Il fournit des informations précieuses sur la contribution relative des différentes régions de fréquences au contenu énergétique global du signal.	$BER_t = \frac{\sum_{n=1}^{F-1} m_t(n)^2}{\sum_{n=F}^{N} m_t(n)^2}$

le Spectral Centroid (SC)	Le SC nous permet d'estimer le "centre de gravité" ou la fréquence moyenne du spectre d'un signal. Il fournit des informations précieuses sur les caractéristiques spectrales et l'équilibre d'un signal.	$SC_t = \frac{\sum_{n=1}^{N} m_t(n) \cdot n}{\sum_{n=1}^{N} m_t(n)}$
Le bande passante (BW)	Le BW nous permet de mesurer la largeur ou la propagation du contenu fréquentiel dans le spectre d'un signal. Il fournit des informations sur la gamme de fréquences présentes dans le signal et peut être calculé de différentes manières	$BW_t = \frac{\sum_{n=1}^{N} \lvert n - SC_t \rvert \cdot m_t(n)}{\sum_{n=1}^{N} m_t(n)}$

Tableau 2 - les Fonctionnalités du domaine fréquentiel

6. Visualisation du son (Spectrogrammes):

La visualisation du son à travers des spectrogrammes implique le processus d'obtention d'une matrice en prenant l'amplitude au carré de la transformée de Fourier à court terme (STFT). Contrairement à la STFT originale, cette matrice comprend des nombres réels au lieu de nombres complexes. Les spectrogrammes ont une importance immense dans les applications audio IA car ils servent de fonctionnalités cruciales introduites dans les algorithmes. Dans les spectrogrammes, l'axe des x représente des temps discrets, englobant toutes les images ou tranches de temps, tandis que l'axe des ordonnées représente la fréquence, incorporant diverses tranches de fréquence. En examinant différentes images dérivées du signal d'origine, nous pouvons observer l'évolution de différentes composantes de fréquence dans le temps. Cette représentation dynamique des changements de fréquence en fonction du temps est la raison pour laquelle le spectrogramme est appelé représentation temps-fréquence, jouant un rôle central dans l'analyse des données audio.

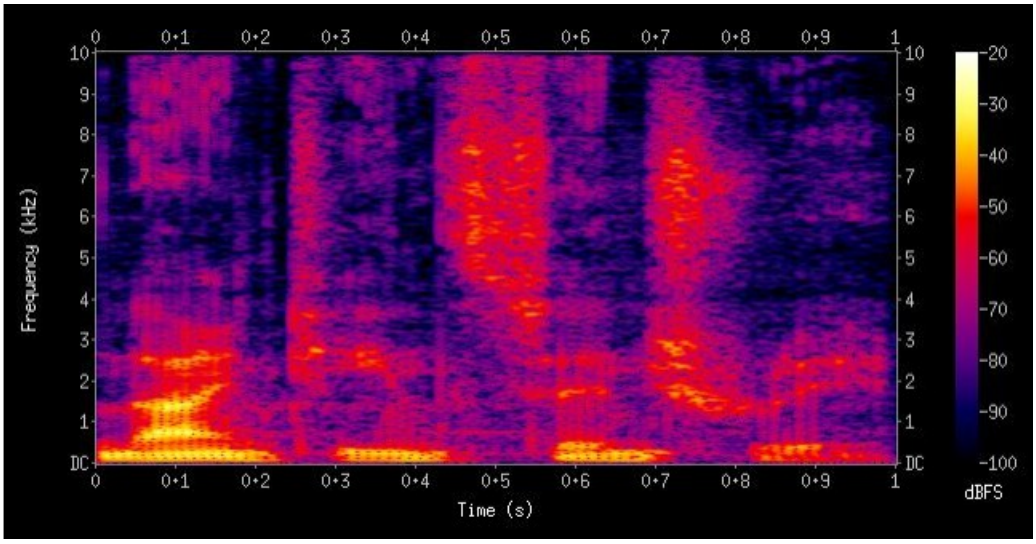


Figure 3: Exemple de Spectrogrammes

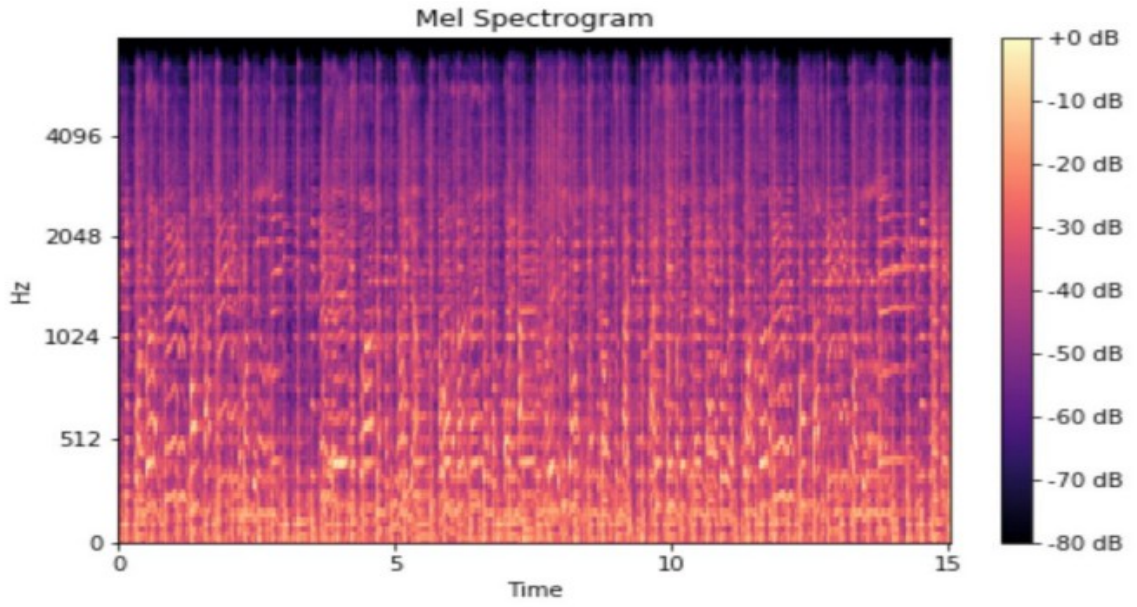


Figure 4: Exemple de Mel Spectrogrammes

7.pipeline d'extraction: pour extraire les caractéristiques de domaine et de fréquence :

A-Pipeline de fonctionnalités à domaine unique: nous partons du son analogique, nous faisons l'ADC (en appliquant l'échantillonnage et la quantification) puis nous obtenons notre version numérisée du son, nous encadrons le signal pour obtenir un tas d'images, maintenant nous calculons les caractéristiques du domaine temporel sur chacune des différentes images puis nous agrégeons ces résultats (en utilisant la moyenne, la médiane, le GMM: les modèles de mélange gaussien [25], ...) puis nous obtenons une caractéristique de valeur ou de vecteur ou de matrice .. pour l'ensemble du son

B-Pipeline de fonctionnalités dans le domaine fréquentiel: nous utilisons la transformée de Fourier pour passer du domaine temporel au domaine fréquentiel, ce qui est similaire au domaine temporel, mais après avoir appliqué l'ADC et le cadrage, nous passons à une étape appelée fenêtrage, ce qui signifie que nous appliquons la fonction de fenêtrage à chaque trame qui éliminera les échantillons aux deux extrémités de la trame et qui génère un signal périodique qui minimise les fuites spéciales, les fuites spéciales sont un problème qui se produit presque tout le temps, lorsque nous prenons la transformée de Fourier d'un signal qui n'est pas un nombre entier de périodes donc le les points finaux sont discontinus, ce discontinu apparaît comme des composants haute fréquence non présents dans le signal d'origine, la fameuse fonction de fenêtrage que nous utilisons le plus souvent est la "fenêtre de Hann"

si nous appliquons une trame non superposée après le fenêtrage, nous perdrons le signal, nous résolvons cela en superposant les trames

maintenant, nous appliquons la transformée de Fourier, il ne reste plus qu'à calculer les caractéristiques et à appliquer l'agrégation et nous obtenons les caractéristiques valeur/vecteur/matrice comme nous l'avons fait dans le pipeline de domaine temporel

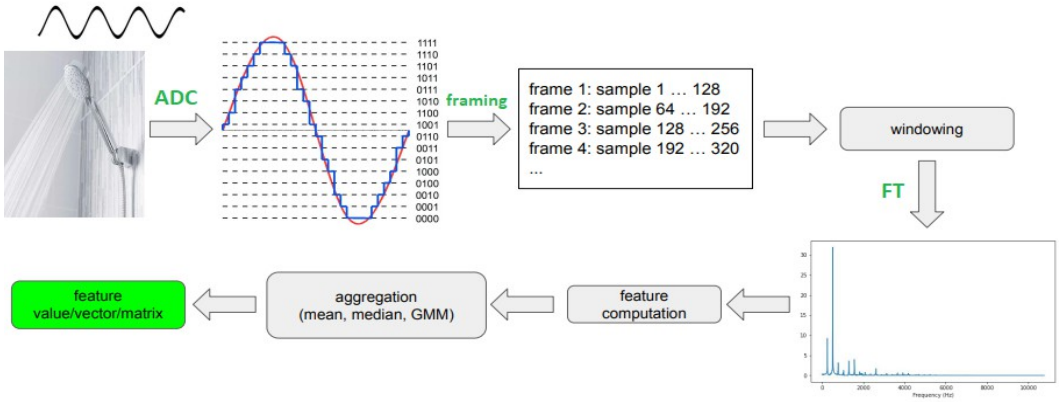


Figure 5: Pipeline de fonctionnalités dans le domaine fréquentiel

Part 2: Traitement de texte

1. Introduction:

Le traitement de texte consiste à manipuler et à analyser des données textuelles de manière structurée et significative. Cela implique une gamme de techniques d'extraction, de nettoyage, de transformation et d'analyse de données textuelles pour découvrir des informations, des modèles et des tendances. Ces techniques sont utilisées dans diverses applications, y compris le traitement du langage naturel (NLP).

2. traitement du langage naturel NLP:

Le traitement du langage naturel (NLP) [29] est un sous-domaine de l'informatique et de l'intelligence artificielle qui se concentre sur l'interaction entre les ordinateurs et le langage humain. La NLP [29] consiste à développer des algorithmes et des programmes informatiques capables de traiter et d'analyser de grandes quantités de données en langage naturel, y compris du texte et de la parole.

L'objectif principal de la NLP est de permettre aux machines de comprendre, d'interpréter et de générer le langage humain. Cela implique un large éventail de tâches, notamment la traduction linguistique, l'analyse des sentiments, la reconnaissance vocale, la réponse aux questions et la synthèse de texte.

La NLP s'appuie sur un large éventail de disciplines, notamment l'informatique, la linguistique, la psychologie et les sciences cognitives. Cela implique l'utilisation de techniques telles que l'apprentissage automatique, l'apprentissage en profondeur et l'analyse statistique pour créer des modèles et des algorithmes capables de traiter et d'analyser des données linguistiques.

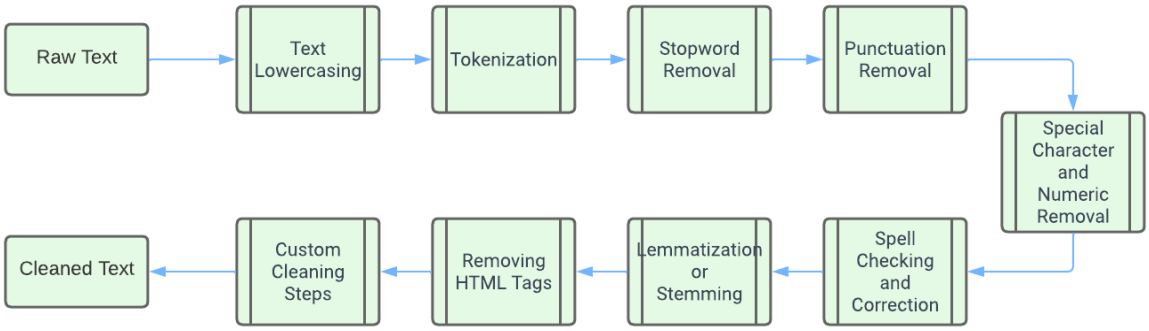


Figure 6: **les étapes de prétraitement pour le nettoyage du texte**

il existe de nombreuses techniques de traitement de texte, les plus utilisées sont:

pré-traitement technique	la description
Texte en minuscules	Convertissez tout le texte en minuscules pour normaliser les données et réduire la taille du vocabulaire.
La tokenisation	est le processus de décomposition d'un morceau de texte en unités plus petites appelées jetons. Ces jetons sont généralement des mots, mais ils peuvent également être des expressions, des phrases ou d'autres unités de langage. les jetons sont segmentés selon certaines règles, telles que les espaces ou la ponctuation.
Les mots vides	sont des mots courants qui sont souvent supprimés du texte lors du prétraitement. Ces mots sont considérés comme moins importants pour comprendre le sens d'une phrase ou d'un document, car ils sont souvent utilisés à des fins grammaticales et n'ont pas beaucoup de sens sémantique,
Suppression de la ponctuation	éliminez les signes de ponctuation du texte car ils ne contribuent souvent pas à la signification sémantique.
Suppression des caractères spéciaux et des chiffres	Supprimez les caractères spéciaux, les symboles et les valeurs numériques qui peuvent ne pas être pertinents pour l'analyse.
Vérification et correction orthographique	Appliquez un algorithme de vérification orthographique pour identifier et corriger les mots mal orthographiés dans le texte.
La radicalisation et la lemmatisation	sont deux techniques utilisées pour transformer les mots en leur forme de base ou racine. -La radicalisation consiste à réduire un mot à sa forme de base en supprimant les suffixes. La racine résultante n'est pas nécessairement un mot valide dans la langue -La lemmatisation consiste à réduire un mot à sa base ou à sa forme de dictionnaire, appelée lemme. Le lemme résultant est toujours un mot valide dans la langue.
Suppression des balises	si le texte contient des balises HTML, supprimez-les pour extraire uniquement le contenu textuel pertinent.

HTML	
Étapes de nettoyage personnalisées	selon les exigences spécifiques de la tâche ou les caractéristiques des données, des étapes de nettoyage supplémentaires peuvent être ajoutées. Celles-ci peuvent inclure la suppression de modèles spécifiques ou de modèles uniques à l'ensemble de données, la gestion du bruit spécifique au domaine ou l'application de règles spécifiques au domaine.

Tableau 3 - prétraitement technique du texte

3. Techniques de vectorisation:

Les techniques de vectorisation en NLP convertissent les données textuelles en représentations numériques, permettant un traitement et une analyse efficaces.

Techniques de vectorisation	la description
TF-IDF (Term Frequency-Inverse Document Frequency)	est une statistique numérique qui reflète l'importance d'un mot dans un document d'une collection ou d'un corpus. Il est calculé en multipliant la fréquence de terme (TF) d'un mot dans un document par la fréquence de document inverse (IDF) du mot à travers le corpus. La fréquence des termes (TF) est une mesure de la fréquence à laquelle un mot apparaît dans un document. Elle est calculée comme le nombre de fois qu'un mot apparaît divisé par le nombre total de mots. Inverse Document Frequency (IDF) est une mesure de la fréquence ou de la rareté d'un mot dans l'ensemble du corpus. Il est calculé comme le logarithme du nombre total de documents dans le corpus, divisé par le nombre de documents contenant le mot.
L'encodage one-hot	One-Hot Encoding is a technique where each word or category is represented as a binary vector, with a value of 1 in the position corresponding to its index and 0 elsewhere.
BOW (Bag-of-Words)	est une technique de vectorisation qui représente le texte en comptant les occurrences de mots dans un document.
CBOW (sac de mots continu)	est un modèle utilisé dans Word2Vec, qui prédit un mot cible en fonction des mots de contexte qui l'entourent. Il représente les mots comme des vecteurs denses, capturant le sens sémantique des mots.
Skip-gram	est un modèle utilisé dans Word2Vec, qui prédit les mots de contexte étant donné un mot cible. Il représente également les mots comme des vecteurs denses, en se concentrant sur la capture des relations entre les mots.
Word2Vec [4]	est une technique populaire d'incorporation de mots qui utilise des modèles CBOW ou Skip-Gram pour créer des représentations de mots denses. Il capture les relations sémantiques entre les mots en fonction de leurs modèles de cooccurrence.
GloVe[2] (Global Vectors for Word Representation)	est une technique d'incorporation de mots qui apprend les représentations de mots sur la base des statistiques globales de cooccurrence de mots. Il considère la matrice globale de cooccurrence de mots du corpus pour générer des vecteurs de mots significatifs.

Tableau 4 - Techniques de vectorisation en NLP

4. Techniques de transformateur:

Transformers have revolutionized (NLP) by capturing contextual information and dependencies in text. Their attention-based architecture enables superior performance in various NLP tasks, making transformers a cornerstone of modern language understanding models. dans notre travail nous nous intéressons plus aux techniques des transformateurs

Techniques de transformateur	la description
Bert (Bidirectional Encoder Representations from Transformers)	est un modèle basé sur un transformateur qui pré-entraîne sur de grandes quantités de données non étiquetées et apprend les intégrations de mots contextualisées. Il capture le contexte bidirectionnel et a été efficace dans diverses tâches NLP.
sBert (Sentence-BERT)	is a modification of BERT that uses siamese and triplet network architectures to learn fixed-length sentence embeddings. It encodes sentences into fixed-dimensional vectors, enabling semantic similarity computations between sentences.

Tableau 5 - Techniques de transformateur en NLP

Part 3: travaux connexes

1. Introduction:

La récupération audio avec des sous-titres écrits par l'homme est un domaine de recherche émergent avec des applications potentielles dans divers domaines. Développer des systèmes de sous-titrage précis et fiables, ainsi que des méthodes efficaces pour faire correspondre les sous-titres avec le contenu audio correspondant, font partie des principaux défis dans ce domaine. Des algorithmes d'apprentissage automatique ont été explorés pour faire correspondre avec précision les sous-titres au contenu audio malgré les variations de langue, de dialecte et de ton, tandis que des techniques de traitement du langage naturel ont été étudiées pour améliorer la précision de la correspondance des sous-titres. Malgré des progrès significatifs ces dernières années, de nombreuses questions et défis de recherche ouverts doivent encore être résolus. Dans les sections suivantes, nous passerons en revue les travaux connexes dans ce domaine et discuterons des derniers développements et des orientations futures de la recherche.

2. Le document 1 [35]: le système de récupération audio basé sur la langue décrit dans le rapport technique se concentre sur la récupération d'un clip audio correspondant à partir d'un groupe de candidats à l'aide d'une requête en langage naturel. Le système se compose d'un encodeur audio et d'un encodeur de texte qui extraient des caractéristiques pour l'audio et le texte, respectivement. L'encodeur audio utilise des réseaux de neurones audio pré-formés (PANN) formés sur l'ensemble de données AudioSet, tandis que l'encodeur de texte utilise BERT (Représentations d'encodeurs bidirectionnels de transformateurs). Les caractéristiques extraites des deux modalités sont ensuite mappées sur un espace d'intégration partagé à des fins de comparaison de similarité.

L'encodeur audio utilise l'architecture ResNet38 ou CNN14 des PANN. Après avoir obtenu la carte des caractéristiques du dernier bloc convolutif, une opération de regroupement

moyen est effectuée le long de la dimension de fréquence, suivie d'opérations maximales et moyennes le long de la dimension temporelle. Les caractéristiques résultantes sont ensuite projetées dans l'espace d'intégration partagé à l'aide d'un bloc de perceptron multicouche (MLP).

Pour l'encodeur de texte, le BERT pré-formé est utilisé pour extraire des représentations de mots contextualisées. Un jeton "CLS" est ajouté au début de chaque phrase, et la sortie de ce jeton est considérée comme la représentation finale de la phrase. Un bloc MLP indépendant est utilisé pour projeter la représentation de la phrase dans l'espace d'intégration partagé.

Pendant l'inférence, le système calcule l'intégration de texte pour la légende interrogée à l'aide de l'encodeur de texte et la compare aux intégrations audio de tous les candidats dans l'ensemble de données, qui sont calculées par l'encodeur audio. Le clip audio avec le score de similarité cosinus le plus élevé est récupéré comme clip correspondant.

Les résultats démontrent la performance supérieure du système proposé par rapport à la ligne de base. Les modèles CNN14+BERT et ResNet38+BERT obtiennent des résultats similaires, indiquant l'efficacité des deux encodeurs audio.

3. Le document 2 [36]: Le système vise à établir des associations significatives entre les clips audio et leurs sous-titres correspondants. Il utilise des réseaux d'intégration audio et de sous-titrage séparés pour mapper les spectrogrammes et les descriptions dans un espace D-dimensionnel partagé. La formation est effectuée par une formation contrastive, améliorant la similarité pour les paires de sous-titres audio correspondantes tout en éliminant les représentations pour les paires non concordantes.

Pour estimer l'accord entre un extrait audio et une description, une matrice de similarité C est calculée en utilisant le produit scalaire normalisé dans l'espace d'intégration partagé. Le système utilise la perte NT-Xent [30], qui fait la moyenne de la perte d'entropie croisée sur les dimensions audio et texte, en utilisant la matrice d'identité I comme vérité fondamentale.

Pour une régularisation et une meilleure généralisation, le système applique plusieurs techniques lors de la formation. Ceux-ci incluent l'amplification du gain pour l'invariance du volume, SpecAugment pour masquer les bandes de temps et de fréquence dans les spectrogrammes et Freq-MixStyle pour transférer les caractéristiques de style de l'appareil.

Dans la phase d'augmentation du texte, le système utilise la rétro-traduction et l'augmentation facile des données. La rétro-traduction introduit des variations en traduisant les phrases dans une langue étrangère et dans la langue source, tandis que l'augmentation facile des données applique des manipulations au niveau des mots telles que l'insertion, la suppression, l'échange et le remplacement.

Le système est formé sur l'ensemble de données ClothoV2.1 [6], avec des extraits audio rembourrés et transformés en spectrogrammes log-MEL à 64 cases. La normalisation par lots est appliquée pour normaliser les caractéristiques audio. Le modèle d'intégration audio, basé sur l'architecture CNN10, agrège les sorties du spectrogramme et subit une transformation à l'aide d'un réseau de neurones. Le modèle d'intégration de texte est basé sur

le modèle BERT et génère des intégrations pour les sous-titres audio. Les modèles sont optimisés conjointement à l'aide de la descente de gradient, avec SMBO utilisé pour l'optimisation des hyperparamètres.

En résumé, le système intègre l'audio et le sous-titrage via des réseaux séparés, utilise une formation contrastive avec perte NT-Xent [30] et utilise des techniques de régularisation telles que gain boost, SpecAugment et Freq-MixStyle. Il s'entraîne sur l'ensemble de données ClothoV2.1 [6], transforme l'audio en spectrogrammes et utilise les architectures CNN10 et BERT pour les modèles d'intégration audio et de texte. La descente de gradient est utilisée pour l'optimisation et SMBO aide au réglage des hyperparamètres.

Les résultats montrent que ce système personnalisé de récupération audio surpasse le système de base et réalise des améliorations significatives. En utilisant le modèle d'intégration de texte BERT plus puissant au lieu de Word2Vec, notre système de base démontre déjà de meilleures performances. De plus, le transfert de poids préformés à partir d'un modèle CNN10 formé sur AudioSet améliore encore le système, ce qui entraîne une amélioration d'environ 10 points de pourcentage de mAP. Ces résultats valident l'efficacité de l'utilisation de techniques d'intégration avancées et de stratégies de préformation pour relever le défi de la rareté des données dans la récupération audio.

4. Le document 3 [37]: Le texte fourni décrit une approche de recherche pour la tâche de récupération audio basée sur la langue du défi DCASE2022. La tâche se concentre sur la recherche d'enregistrements audio qui correspondent étroitement à une requête textuelle donnée, avec des applications potentielles dans les moteurs de recherche pour les fichiers audio basés sur des descriptions textuelles de forme libre. L'approche décrite dans le texte s'appuie sur des modèles pré-entraînés et un cadre d'apprentissage métrique pour lier sémantiquement les modalités audio et textuelles.

Le système se compose de deux composants : une tour audio et une tour de texte. Chaque tour traite les données d'entrée respectives séparément. La tour audio utilise un encodeur audio pré-entraîné, tandis que la tour de texte utilise un encodeur de texte pré-entraîné. Ces encodeurs génèrent des représentations intermédiaires ou des intégrations pour les entrées audio et texte. Les intégrations sont ensuite traitées par des adaptateurs pour aligner leur dimensionnalité et permettre des comparaisons. Les adaptateurs sont des architectures de réseaux neuronaux superficiels. L'alignement est effectué à l'aide de techniques d'apprentissage métrique, où les exemples positifs (paires audio-texte similaires) sont encouragés à se rapprocher dans l'espace d'intégration, tandis que les exemples négatifs (paires audio-texte dissemblables) sont écartés.

Deux fonctions de perte différentes sont explorées dans les expériences : la perte contrastive et la perte d'entropie croisée à l'échelle de la température normalisée (NT-Xent).

La perte contrastive mesure la similarité entre les plongements basés sur la similarité cosinus, tandis que la perte NT-Xent est utilisée par les principales soumissions du défi.

Les expériences menées dans l'étude utilisent l'ensemble de données de développement Clotho v2 et étendent les données de formation avec des paires texte et audio faiblement alignées collectées à partir de la plate-forme Freesound. Pour plus de simplicité et de reproductibilité, nous nous limitons aux sous-ensemble dev de l'ensemble de données

FSD50k [31]. L'évaluation des performances du système est basée sur des mesures de rappel et de précision moyenne moyenne.

Les détails de mise en œuvre incluent l'utilisation du framework PyTorch [11], de la bibliothèque Transformers [33] pour le traitement de texte, du modèle de base distilroberta pré-entraîné (une version compressée du modèle RoBERTa original [34]) comme encodeur de texte et des PANN pré-entraînés [20] comme encodeur audio. Des réseaux de neurones à réaction simple sont utilisés comme adaptateurs.

Le document mentionne également la soumission de quatre configurations de système différentes, et les résultats des systèmes soumis sont comparés à la ligne de base du défi et aux équipes les mieux classées.

Les résultats montrent que cette approche permet d'obtenir des résultats de bonne qualité dans la configuration de formation standard, avec des performances compétitives par rapport à la ligne de base du défi et aux principales soumissions. Cependant, l'inclusion de données d'entraînement bruyantes supplémentaires a un impact négatif sur les performances de récupération. Le réglage fin des modèles en fonction de l'ensemble de données du défi améliore encore les résultats. Ces résultats mettent en évidence l'efficacité de notre approche dans la récupération audio basée sur la langue et l'importance d'une sélection minutieuse des données et de l'adaptation du modèle pour des performances optimales.

5. **Recapitalisation:**

Un tableau de la récapitulation des 3 travaux connexes:

travaux connexe	Audio feutures	Text feutures	Model utiliser	Data sets	Avantage	Désavantage
Le document 1 [35]:	Log-mel spectrogram avec PANNs	BERT	Audio encoder and a text encoder	Clotho V2.1 et AudioSet dataset	- cette méthode améliore significativement toutes les métriques par rapport au système de référence.	-Le modèle de pré-formation sur AudioCaps [5] n'améliore pas les performances du système sur l'ensemble de données Clotho.
Le document 2 [36]	Log-mel spectrogram	BERT model (’bert-base-uncased’)	CNN10 embedding model	Clotho V2.1 [6]	-l'utilisation de modèles d'intégration audio et de texte pré-formés augmente considérablement les performances de récupération sur ClothoV2 -Il a été démontré qu'il a été démontré que l'augmentation des entrées textuelles et audio réduisait considérablement le surajustement	-Améliorations non significatives de la pré-formation sur AudioCaps [5]
Le document 3 [37]	Log-mel spectrogram avec pré-formé	pretrained distilroberta-base model	simple feed-forward neural	Clotho v2, dev subset de FSD50k	- résultats prometteurs lors du pré-entraînement des modèles avec des	-le choix de la fonction de perte montre une grande partie de

	PANNs model		networks (NN)	[31]	données bruitées	l'écart de performance -encodeurs fixes au lieu d'un réglage fin

Tableau 6 - récapitulation des travaux connexes

6. Conclusion: ce chapitre a fourni une exploration complète du traitement audio, du traitement de texte et des travaux connexes. Le chapitre commençait par une introduction, préparant le terrain pour les discussions ultérieures. La partie 1 s'est concentrée sur le travail de traitement audio, tandis que la partie 2 s'est penchée sur le travail de traitement de texte dans le domaine du traitement du langage naturel (NLP). Enfin, la partie 3 a présenté divers travaux liés à l'emploi dans le domaine. Dans l'ensemble, ce chapitre a jeté des bases solides pour une enquête plus approfondie dans le monde fascinant du traitement audio et de texte.

Chapitre 2: Approche proposée

1. Introduction:

La récupération audio avec des légendes écrites humaines est un processus qui implique plusieurs étapes pour permettre aux utilisateurs de rechercher et de récupérer un contenu audio spécifique à l'aide de descriptions écrites. Dans ce processus, nous commençons par un texte écrit, qui est utilisé comme requête de recherche pour récupérer le contenu audio pertinent.

La première étape consiste à extraire les caractéristiques du texte écrit, ce qui peut être réalisé grâce à des techniques de traitement du langage naturel. Ces fonctionnalités peuvent inclure des mots-clés, des entités et d'autres informations pertinentes qui peuvent être utilisées pour identifier et récupérer le contenu audio qui correspond à la requête de recherche.

Une fois les caractéristiques textuelles extraites, elles sont transmises à un modèle qui est formé pour convertir ces caractéristiques en caractéristiques audio. Ceci est réalisé en mappant les caractéristiques du texte aux caractéristiques audio, telles que la hauteur, la tonalité et le rythme, qui sont spécifiques au contenu audio.

Une fois les caractéristiques audio extraites, elles sont utilisées pour synthétiser le contenu audio correspondant à la requête textuelle écrite. Ce processus implique la conversion des fonctionnalités audio en signaux audio pouvant être lus via des haut-parleurs ou des écouteurs, permettant aux utilisateurs d'entendre le contenu audio qui correspond à leur requête de recherche.

Dans l'ensemble, la récupération audio avec des légendes écrites humaines est un outil puissant qui peut aider les utilisateurs à trouver et à accéder rapidement au contenu audio pertinent à l'aide de descriptions écrites. En extrayant des caractéristiques du texte et en les mappant à des caractéristiques audio, ce processus permet aux utilisateurs de rechercher et de récupérer du contenu audio de manière plus efficace et accessible.

2. Architecture globale:

L'architecture globale de la récupération audio avec des sous-titres écrits par l'homme intègre plusieurs composants pour permettre aux utilisateurs de rechercher et de récupérer efficacement un contenu audio spécifique à l'aide de descriptions écrites.

Nous avons essayé deux versions d'architectures, la première prend les embeddings audio en entrée pour sortir les embeddings audio, la seconde prend les embeddings de texte pour sortir les embeddings audio

Méthode 1 (Encoder-Encoder): dans la méthode 1, Le module de traitement de texte extrait les caractéristiques pertinentes des légendes écrites ou des requêtes. Cela implique l'utilisation de modèles de langage pré-formés tels que BERT et sBERT pour extraire des caractéristiques significatives du texte.Simultanément, le module de traitement audio se concentre sur l'extraction des caractéristiques du contenu audio qui s'alignent sur les sous-titres écrits. Cela

implique généralement la conversion des échantillons audio en spectrogrammes log-mel, une représentation couramment utilisée dans les modèles d'apprentissage en profondeur pour les données audio.

Les caractéristiques audio et les caractéristiques de texte sont combinées et entrées dans les modèles finaux responsables de l'exécution de la tâche de récupération audio. Les modèles finaux utilisent des architectures CNN [18] ou RNN [17] et sont entraînés sur des spectrogrammes log mel. De plus, le score pertinent entre un signal audio et une description textuelle est calculé en prenant le produit scalaire de leurs incorporations audio et incorporations de texte, qui sont des représentations vectorielles des caractéristiques audio et textuelles.L'architecture résultante se compose de quatre modèles distincts :

Model 1: nous passons spectrogramme log mel en entrée de CNN et calcul du score pertinent avec les incorporations de sBert

Model 2: nous passons spectrogramme log mel en entrée de RNN et calcul du score pertinent avec les incorporations de sBert

Model 3: nous passons spectrogramme log mel en entrée de CNN et calcul du score pertinent avec les incorporations de Bert

Model 4: nous passons spectrogramme log mel en entrée de RNN et calcul du score pertinent avec les incorporations de Bert

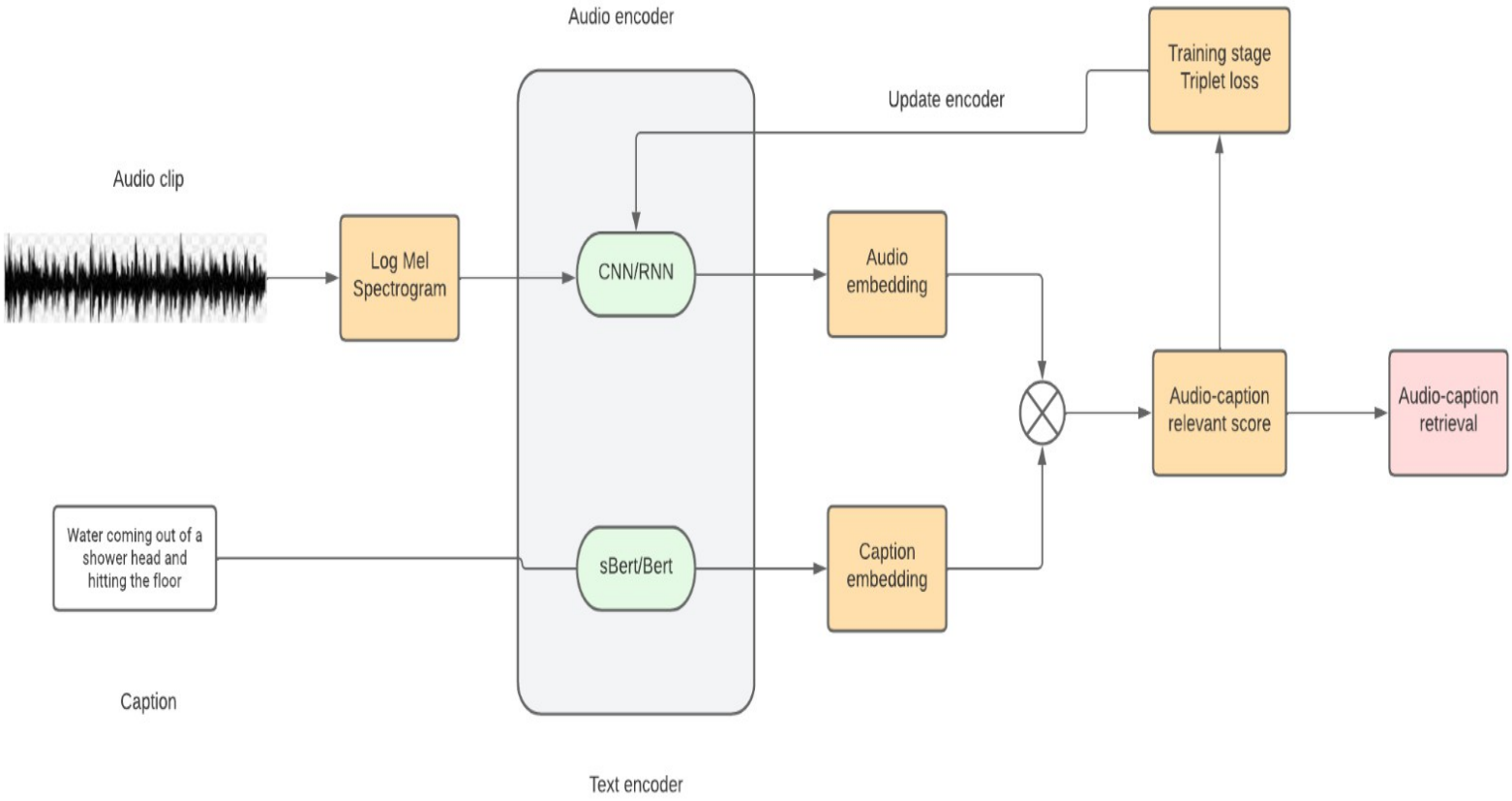


Figure 7: L'architecture du système pour la tâche de « moteur de recherche audio basé sur des requêtes textuelles ».

Méthode 2 (Encoder-Encoder): dans la méthode 2, nous utilisons les mêmes étapes en appliquant sbert aux phrases de texte, nous extrayons également les spectrogrammes log mel de la légende audio, puis nous passons l'intégration de texte en entrée au modèle CNN ou RNN pour obtenir l'intégration audio en sortie, L'architecture résultante consiste de deux modèles distincts :

Model 5: nous passons l'intégration de sbert comme entrée à CNN pour produire un spectrogramme log mel, nous calculons le score pertinent avec la similarité (similarité cosinus [26]) entre l'audio prédit et l'audio vrai

Model 6: nous passons l'intégration de sbert comme entrée à RNN pour produire un spectrogramme log mel, nous calculons le score pertinent avec la similarité (similarité cosinus [26]) entre l'audio prédit et l'audio vrai

3. Architecture détaillée:

Maintenant que nous avons une compréhension de haut niveau de l'architecture globale de la récupération audio avec des sous-titres écrits par l'homme, plongeons dans l'architecture détaillée de chaque composant. Dans cette section, nous aborderons plus en détail le module de traitement de texte, le module de traitement audio et les modèles finaux, y compris les modèles de réseau neuronal spécifiques et les techniques utilisées dans chaque module. En comprenant l'architecture détaillée de chaque composant, nous pouvons mieux comprendre comment fonctionne la récupération audio avec des sous-titres écrits par l'homme et comment nous pouvons améliorer les performances du système.

a. caractéristiques du texte (Bert & sBert):

Bert [20]**:** est un modèle de langage basé sur un transformateur préformé qui est largement utilisé pour diverses tâches de traitement du langage naturel (NLP). Le modèle est formé sur un vaste corpus de données textuelles, y compris des articles de Wikipédia, et est capable de générer des incorporations contextualisées riches qui capturent le sens et les relations entre les mots dans une phrase. Dans le contexte de la récupération audio avec des sous-titres écrits par l'homme, Bert peut être utilisé pour extraire les caractéristiques pertinentes des sous-titres écrits ou des requêtes. Dans notre cas, nous prenons un ensemble de phrases d'un fichier de sous-titres, qui contient 5 phrases différentes pour chaque échantillon audio, chaque phrase représentant une partie différente de cet échantillon audio. Nous pouvons ensuite saisir ces phrases dans Bert et obtenir un ensemble d'incorporations contextualisées qui capturent le sens sous-jacent de chaque phrase. Le tableau de forme résultant (X,20,768) où X est le nombre de phrases, 20 est le nombre maximum de mots qui existent dans chaque phrase et 768 est la forme de chaque mot. Cependant, il convient de noter que l'utilisation de Bert pour la récupération audio avec des sous-titres écrits par l'homme peut nécessiter plus de RAM et de temps en raison de sa forme de tableau plus élevée. Il a été démontré que Bert surpasse les autres modèles de langage dans diverses tâches NLP, ce qui en fait un outil puissant. pour la récupération audio avec des légendes écrites par l'homme.

Sbert [19]**:** d'autre part, est une version modifiée de Bert qui a été spécialement conçue pour

les incorporations au niveau de la phrase. Contrairement au Bert original, qui génère des intégrations au niveau du mot, sBert génère des intégrations au niveau de la phrase. Cela fait de sBert un meilleur choix pour les tâches qui impliquent de comparer et de faire correspondre des phrases. Dans le contexte de la récupération audio avec des sous-titres écrits par l'homme, sBert peut être utilisé pour extraire des incorporations au niveau de la phrase à partir des sous-titres écrits ou des requêtes. Dans notre cas, nous pouvons utiliser le même fichier de sous-titres et entrer chaque phrase dans sBert pour obtenir un ensemble d'incorporations qui capturent le sens de chaque phrase. Le tableau de forme résultant (X,768) où X est le nombre de phrases et 768 est la forme de la phrase entière. Il convient de noter que si deux mots sont écrits de la même manière mais ont des significations différentes, sBert les traitera comme des mots différents, contrairement à Bert où ils sont considérés comme identiques. Cela fait de sBert un outil précieux pour la récupération audio avec des sous-titres écrits par l'homme, en particulier lorsque l'accent est mis sur la sémantique au niveau de la phrase. analyse détaillée de mots ou d'expressions individuels dans une phrase.

b. stockage du son sur la base de données:

Le stockage du son sur des bases de données joue un rôle crucial dans la gestion et l'exploitation efficaces de la puissance des données audio dans diverses industries et applications. Avec l'importance croissante du contenu audio dans le monde actuel axé sur les données, des mécanismes de stockage et de récupération efficaces sont essentiels. Le processus consiste à collecter des échantillons audio provenant de diverses sources et à les organiser méticuleusement dans la base de données avec les métadonnées pertinentes. Ces métadonnées, y compris les détails d'enregistrement et les informations contextuelles, enrichissent le contenu audio et contribuent à une gestion efficace. Les bases de données offrent des fonctionnalités précieuses telles que l'indexation et la catégorisation, facilitant le stockage et la récupération efficaces des sons. L'indexation basée sur le genre, l'artiste, l'instrument ou d'autres attributs permet aux utilisateurs de rechercher rapidement et précisément un contenu audio spécifique. La protection des données est une considération importante et les bases de données fournissent des mécanismes robustes pour garantir l'intégrité et la sécurité des fichiers audio, y compris des options de sauvegarde et de récupération. Ces mesures protègent contre les pertes ou la corruption potentielles, instillant la confiance dans les organisations fortement dépendantes des actifs audio. L'exploitation des capacités des bases de données permet en outre des capacités de requête avancées, permettant aux utilisateurs d'effectuer des recherches complexes et de récupérer des fichiers audio précis. en combinant divers critères de recherche, tels que le genre, l'artiste et l'année, les utilisateurs peuvent localiser efficacement un contenu audio spécifique dans la base de données. Dans l'ensemble, l'utilisation de bases de données pour le stockage du son offre une solution structurée et évolutive, optimisant la gestion et l'utilisation des actifs audio dans le monde centré sur les données d'aujourd'hui.

c. Modèle (CNN, RNN):

Le choix de l'architecture du modèle joue un rôle crucial dans la réussite de tout projet de deep learning. Dans notre cas, nous avons exploré deux architectures populaires, le réseau neuronal convolutif (CNN) et le réseau neuronal récurrent (RNN), qui ont tous deux leurs avantages et leurs limites uniques. Le CNN est un choix populaire pour les tâches de

traitement d'images et de la parole, tandis que le RNN est souvent utilisé pour les données séquentielles, telles que le texte et les données de séries chronologiques. Chaque architecture a ses spécificités, et choisir la bonne nécessite une compréhension du domaine du problème et des données disponibles. Dans les sections suivantes, nous approfondirons les architectures CNN et RNN utilisées dans notre projet, leurs choix de conception et la façon dont elles sont formées pour atteindre des performances élevées sur notre tâche.

1- **CNN:**

1.1 Introduction:

Est un type spécialisé de réseau de neurones qui excelle dans le traitement d'images. Ils surpassent les autres architectures de réseaux neuronaux, telles que la perception multicouche, en raison de leur capacité à extraire des caractéristiques à partir de données d'image structurées. Les CNN ont moins de paramètres que les couches denses et sont mieux adaptés au traitement des images. L'intuition derrière les CNN est basée sur le système de vision humaine, où les composants du réseau apprennent à extraire différentes caractéristiques. Les deux composants principaux d'un CNN sont la convolution et la mise en commun.

La convolution est le processus d'application d'un noyau ou d'un filtre à une image. Le noyau est une grille de pondérations appliquée à l'image, ce qui donne une grille de sortie de la même taille que l'image d'origine. Pour calculer la sortie, le noyau est centré sur l'image et la valeur de convolution est le produit scalaire entre les deux vecteurs de l'image et le noyau. Le noyau est un détecteur de caractéristiques et ses valeurs sont apprises au cours du processus de formation. Les décisions architecturales pour la convolution incluent la taille de la grille, la foulée, la profondeur et le nombre de noyaux.

La mise en commun est le processus de sous-échantillonnage d'une image. La mise en commun maximale est le type de mise en commun le plus courant dans DL, où une grille est superposée à l'image et la valeur maximale est sélectionnée. La mise en commun moyenne est une autre option où la valeur moyenne est prise. La mise en commun réduit la taille de l'image et simplifie la sortie pour les couches suivantes.

L'architecture CNN commence par la couche d'entrée et passe à la phase d'apprentissage des fonctionnalités où plusieurs couches de convolution sont suivies de couches de regroupement. Au fur et à mesure que nous passons d'une couche de convolution à la suivante, des caractéristiques de niveau supérieur sont extraites. Les résultats sont aplatis et une couche entièrement connectée traite le vecteur 1D. À la fin, un classificateur de fonction fournit une distribution de probabilité pour différentes catégories.

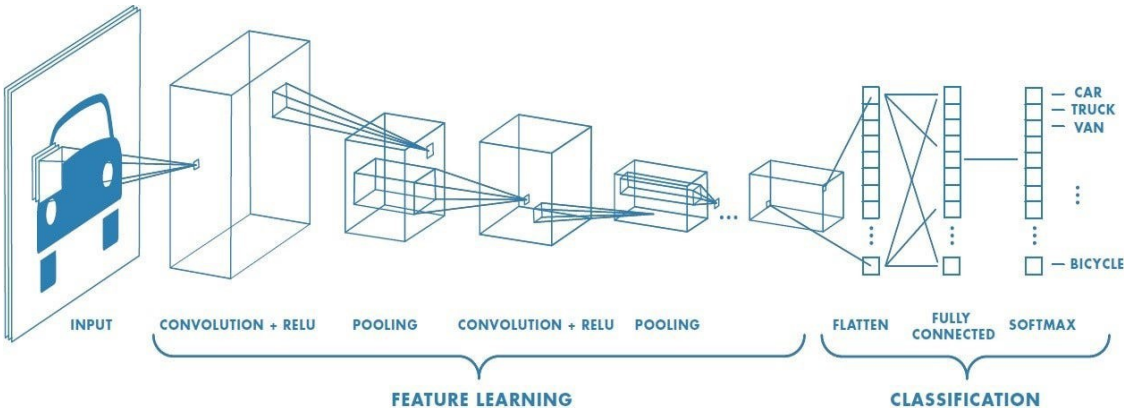


Figure 8: les étapes de l'application de CNN

1.2 Notre définition de l'architecture CNN:

Pour la méthode 1(Encoder-Encoder): Dans ma définition d'architecture pour un réseau de neurones convolutifs (CNN) utilisant la bibliothèque Keras. Le réseau se compose de plusieurs couches qui sont empilées pour extraire les caractéristiques de l'audio d'entrée et faire des prédictions.

La première partie de l'architecture se concentre sur les couches convolutionnelles. Les couches Conv2D effectuent des opérations convolutives sur les données d'entrée, en utilisant un noyau 1x1 et la fonction d'activation ReLU [10]. Le nombre de filtres augmente progressivement de 64 à 192 dans les couches suivantes. Chaque couche convolutive est suivie d'une couche MaxPooling2D, qui réduit les dimensions spatiales des cartes d'entités en effectuant un sous échantillonnage. Une couche BatchNormalization est également incluse après chaque couche convolutive pour normaliser les activations et améliorer la stabilité de l'entraînement.

Après les couches convolutionnelles, l'architecture comprend une couche Flatten pour convertir les cartes d'entités 2D en un vecteur 1D. Cela prépare les données pour les couches ultérieures entièrement connectées (dense).

La partie entièrement connectée de l'architecture se compose de trois couches denses de tailles croissantes : 256, 512 et 1024 unités, respectivement. Chaque couche dense utilise la fonction d'activation ReLU [10] et comprend une couche de décrochage avec un taux de décrochage de 0,2. Les couches d'abandon aident à éviter le surajustement en définissant de manière aléatoire une fraction des unités d'entrée sur 0 pendant l'entraînement.

La couche finale est une couche dense avec 768 unités pour le cas du sBert et 20*768 unités pour le cas du Bert, et une fonction d'activation linéaire. Cette couche produit les prédictions du modèle. La fonction triplet_loss [22] est spécifiée comme fonction de perte lors de la compilation, et l'optimiseur Adam [28] est utilisé avec un taux d'apprentissage de 0,001.

L'architecture est formée en utilisant une taille de lot de 55(batch size) et est formée pour 300 époques pour le sbert model et 50 époques pour le bert model. Le résumé du modèle montre que le nombre total de paramètres dans le réseau dans le cas de sbert est de 2 390 336. Parmi ceux-ci, 2 388 800 paramètres peuvent être entraînés, tandis que les 1 536 paramètres

restants ne peuvent pas être entraînés, alors que dans le cas de bert, il s'agit de 17 347 136. Parmi ceux-ci, 17 345 600 paramètres peuvent être entraînés, tandis que les 1 536 paramètres restants ne peuvent pas être entraînés. Les paramètres pouvant être formés sont ajustés pendant le processus de formation pour minimiser la perte et améliorer la précision du modèle.

Dans l'ensemble, cette architecture combine des couches convolutives et denses pour construire un modèle d'apprentissage en profondeur capable d'extraire des caractéristiques audio à partir de signaux audio et de faire des prédictions basées sur ces caractéristiques. Il offre une flexibilité pour une personnalisation plus poussée et peut être formé à l'aide de la fonction de perte et de l'optimiseur spécifiés.

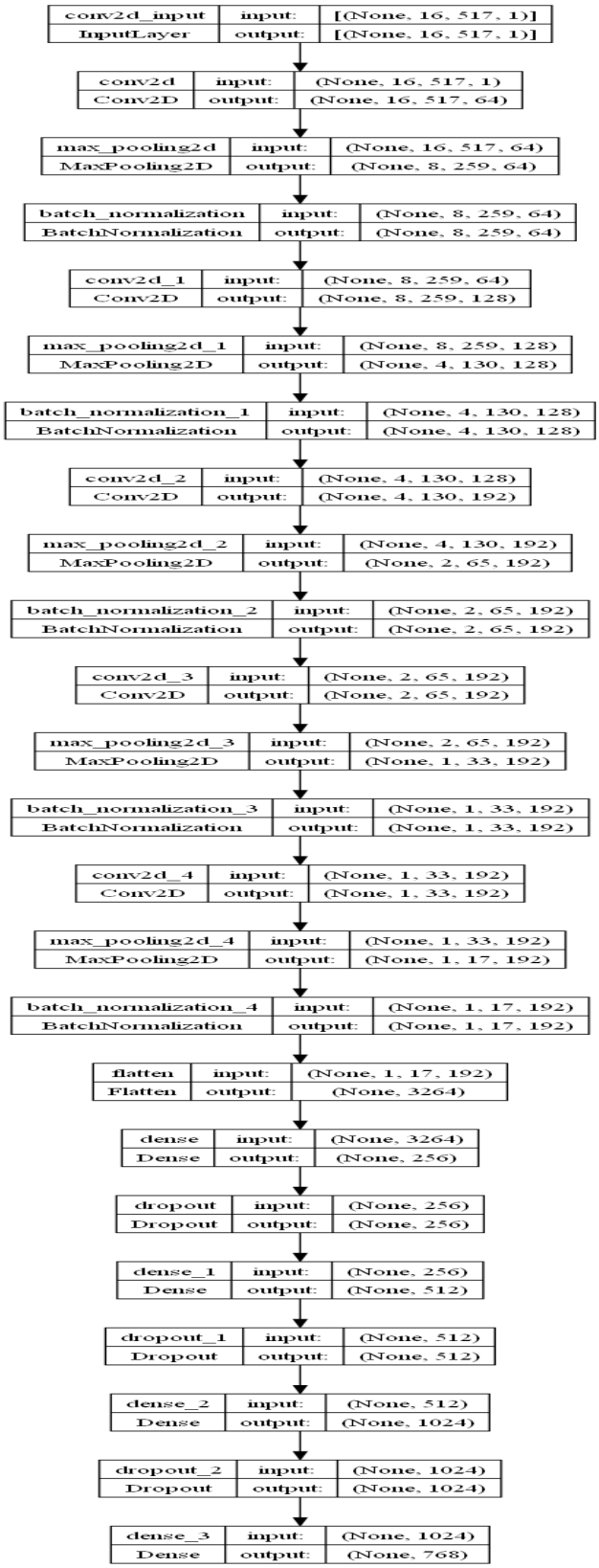


Figure 9: schéma de l'architecture CNN de la méthode 1

Pour la méthode 2 (Encoder-Decoder): L'architecture CNN se compose de plusieurs couches convolutives suivies de couches de mise en commun maximale et de normalisation par lots. La forme d'entrée du réseau est définie et le modèle séquentiel est initialisé. La première couche convolutionnelle a 32 filtres avec une taille de noyau de (1, 1) et utilise la fonction d'activation ReLU [10]. Elle est suivie d'une couche de regroupement maximum avec une taille de pool de (2, 2) et des foulées de (2, 2), ainsi qu'une couche de normalisation par lots.

La deuxième couche convolutionnelle a 64 filtres avec une taille de noyau de (1, 1) et utilise également la fonction d'activation ReLU [10]. Elle est suivie d'une autre couche de mise en commun maximale et d'une couche de normalisation par lots. La troisième couche convolutive a 128 filtres et la quatrième couche convolutive a 192 filtres, tous deux utilisant la fonction d'activation ReLU[10]. Chacune de ces couches convolutionnelles est suivie d'une couche de regroupement maximum et d'une couche de normalisation par lots.

La sortie de la dernière couche convolutive est aplatie et introduite dans deux couches denses. La première couche dense a 256 unités avec la fonction d'activation ReLU et un terme de régularisation de L2 avec un coefficient de 0,001. Une couche de décrochage avec un taux de 0,2 est appliquée après la première couche dense. La deuxième couche dense a 512 unités avec la fonction d'activation ReLU et la même régularisation et abandon L2 que la couche précédente.

La couche de sortie finale a 8272 unités avec une fonction d'activation linéaire. Le modèle est compilé avec l'optimiseur Adam [28], un taux d'apprentissage de 0,001 et une erreur quadratique moyenne (MSE) [27] comme fonction de perte. La métrique de précision est également spécifiée. Le récapitulatif du modèle affiche le nombre total de paramètres, de paramètres entraînables et de paramètres non entraînables :Nombre total de paramètres : 6 771 600, Paramètres pouvant être entraînés : 6 770 768, Paramètres non entraînables : 832.

Le modèle est ensuite entraîné à l'aide de la fonction fit(), avec les tableaux d'entrée spécifiés pour les données d'entraînement et les données de validation. La taille du lot est définie sur 32 et le modèle est entraîné pendant 100 époques avec un brassage des données d'entraînement à chaque époque.

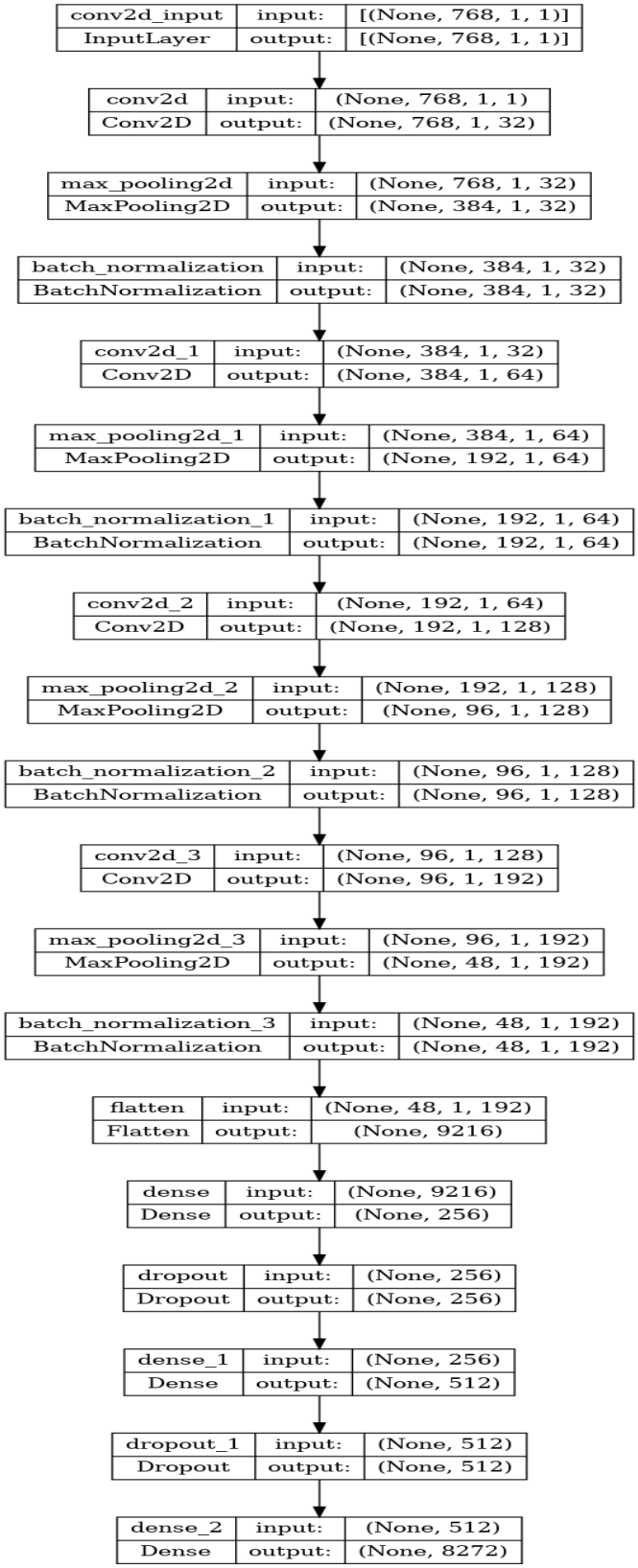


Figure 10: schéma de l'architecture CNN de la méthode 2

2- **RNN:**

2.1 Introduction:

Est un outil essentiel pour le traitement de données séquentielles où l'ordre des éléments est important. Le réseau est capable de traiter des données quel que soit le nombre de mots ou de points de données, et chaque élément est traité dans son contexte. Les RNN fonctionnent particulièrement bien avec l'audio/la musique, où la forme d'onde est une série temporelle univariée, ce qui signifie que nous n'avons qu'une seule mesure prise à chaque intervalle, et les coefficients cepstraux Mel-Frequency (MFCC) sont une série temporelle multivariée. L'architecture RNN se compose d'une couche récurrente, d'une couche dense pour la classification et d'une cellule mémoire qui traite toutes les informations. Le composant le plus important de la couche récurrente est la cellule, qui produit deux choses: le vecteur d'état qui représente la mémoire à un certain moment et qui sera réutilisé à l'étape suivante, et la sortie réelle.

L'entrée du RNN est tridimensionnelle, la première étant la taille du lot, la seconde étant le nombre d'étapes et la troisième étant le nombre de dimensions. La cellule mémoire d'un RNN simple est un réseau de neurones très basique qui utilise une couche dense avec la fonction d'activation « tanh » pour éviter le problème de la disparition et de l'explosion des gradients. La rétropropagation dans le temps (BPTT) est utilisée pour former un RNN, qui rétropropage l'erreur dans le temps. Cependant, les RNN ont une mémoire à long terme limitée, ce qui les rend inadaptés à l'apprentissage à partir de modèles avec de longues dépendances, telles que les données de séries chronologiques, les données audio ou les données musicales. Pour résoudre ce problème, des réseaux de mémoire à long terme LSTM ont été conçus et se sont avérés efficaces.

2.2 LSTM:

Sont un type spécial de réseaux de neurones récurrents (RNN) conçus pour apprendre des modèles à long terme. Contrairement aux RNN traditionnels, les réseaux LSTM utilisent une cellule mémoire pour suivre les informations à long terme. Alors que les RNN ne peuvent détecter que des modèles jusqu'à 100 étapes, les réseaux LSTM peuvent gérer des centaines ou des milliers d'étapes, ce qui les rend idéaux pour les tâches qui impliquent des séquences de données avec des dépendances à long terme.

L'architecture des réseaux LSTM est similaire à celle des RNN simples, à l'exception de la cellule elle-même. Dans LSTM, il existe des composants supplémentaires appelés portes qui agissent comme des filtres d'informations. Ces portes comprennent la porte d'oubli, la porte d'entrée et la porte de sortie. La porte d'oubli aide le réseau à décider quelles informations oublier, tandis que la porte d'entrée décide quelles informations mémoriser.

La cellule LSTM a deux vecteurs d'état - l'état caché et l'état de la cellule. L'état de la cellule est responsable du stockage des informations de la mémoire à long terme, tandis que l'état caché est utilisé à la fois comme sortie et comme nouvelle entrée pour le pas de temps suivant. La porte d'oubli et la porte d'entrée fonctionnent ensemble pour mettre à jour l'état de la cellule. La porte d'oubli utilise une fonction sigmoïde pour déterminer les informations à oublier, tandis que la porte d'entrée utilise une fonction sigmoïde et une fonction tanh pour

décider quelles informations ajouter à l'état de la cellule.

La porte de sortie, qui est également connectée à une fonction sigmoïde et à une fonction tanh, contrôle les informations à sortir de la cellule. Cette sortie est utilisée à la fois comme état caché et comme sortie de la cellule LSTM. Cela permet au réseau de mémoriser ou d'oublier sélectivement des informations selon les besoins, ce qui le rend idéal pour des tâches telles que la reconnaissance vocale, la traduction linguistique et le sous-titrage d'images.

Il existe également d'autres variantes de LSTM, notamment la Gated Recurrent Unit (GRU), qui est une version plus simple de LSTM qui utilise moins de portes. Cependant, LSTM reste le type de RNN le plus largement utilisé en raison de sa capacité à gérer les dépendances à long terme et de son efficacité dans de nombreuses applications.

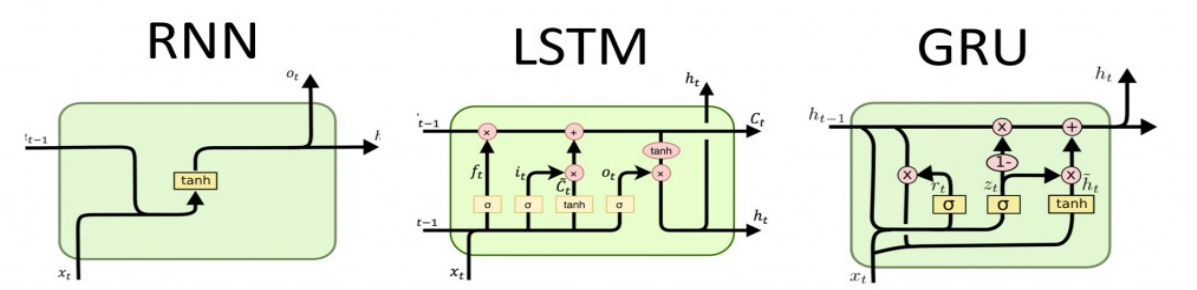


Figure 11: RNN vs LSTM vs GRU

2.3 Notre définition de l'architecture RNN:

Pour la méthode 1: Dans ma définition d'architecture pour un réseau de neurones récurrent (RNN) avec des cellules de mémoire longue à court terme (LSTM) utilisant la bibliothèque Keras. Ce réseau est conçu pour traiter des données séquentielles, telles que des séries chronologiques ou du texte, et faire des prédictions basées sur les modèles appris.

L'architecture commence par la définition séquentielle du modèle. La forme d'entrée est spécifiée pour s'adapter aux séquences de forme (64, 768) pour sbert et (64, 768*20) pour bert, où 64 représente la longueur de la séquence et 768 (ou 768*20) correspond à la dimensionnalité de chaque élément dans la séquence.

Le composant principal de l'architecture est les couches LSTM. La première couche LSTM est ajoutée avec 64 unités et est configurée pour renvoyer des séquences. Cela signifie que la sortie de cette couche est transmise en entrée à la couche suivante tout en préservant la structure de la séquence.

Deux autres couches LSTM suivent, chacune avec un nombre accru d'unités : 128 et 192, respectivement. Toutes ces couches LSTM sont également configurées pour renvoyer des séquences.

Après les couches LSTM, une couche Flatten est incluse pour convertir les données de séquence en un vecteur 1D. Cela prépare les données pour les couches ultérieures entièrement connectées (dense).

Semblable à l'architecture précédente, la partie entièrement connectée se compose de trois couches denses avec respectivement 256, 512 et 1024 unités. Chaque couche dense utilise la fonction d'activation ReLU et comprend une couche de décrochage avec un taux de décrochage de 0,2.

La couche finale est une couche dense avec 768 unités en cas sbert et 768*20 en cas bert, et une fonction d'activation linéaire. Cette couche produit les prédictions du modèle.

L'architecture est formée en utilisant une taille de lot de 55(batch size) et est formée pour 300 époques pour le sbert model et 50 époques pour le bert model. Le résumé du modèle montre que pour le modèle Sbert, le nombre total de paramètres dans le réseau est de 3 316 480. Tous ces paramètres sont entraînables, et il n'y a pas de paramètres non entraînables. et pour le modèle bert, le nombre total de paramètres dans le réseau est 17,681,920. Tous ces paramètres sont entraînables, et il n'y a pas de paramètres non entraînables. Ces paramètres entraînables sont ajustés pendant le processus d'entraînement pour minimiser la perte et améliorer les performances du modèle.

En résumé, cette architecture exploite les couches LSTM pour capturer des modèles séquentiels dans les données d'entrée. Il combine des couches LSTM avec des couches entièrement connectées pour extraire les caractéristiques des séquences et faire des prédictions basées sur ces caractéristiques. L'architecture est flexible et peut être personnalisée davantage.

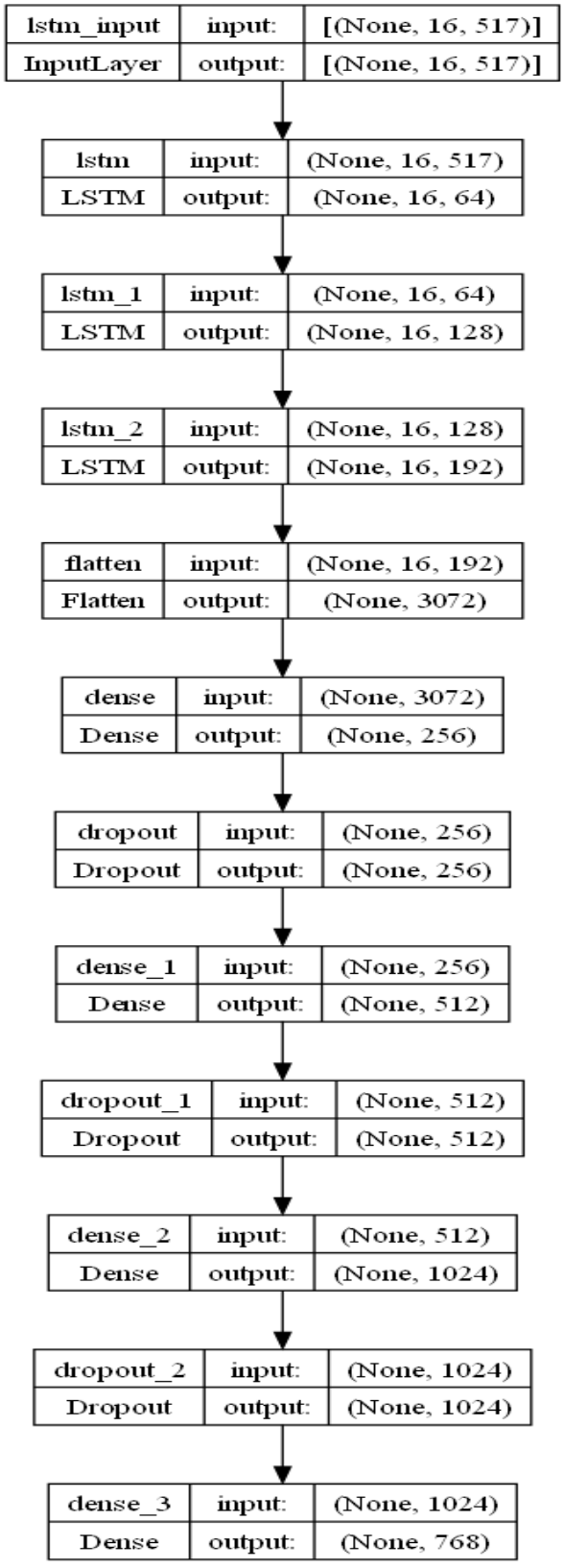


Figure 12: schéma de l'architecture RNN de la méthode 1

Pour la méthode 2: L'architecture RNN se compose de plusieurs couches LSTM (Long Short-Term Memory), qui sont un type de couche récurrente qui peut capturer des dépendances à long terme dans des données séquentielles. La forme d'entrée du modèle est (64, 768), indiquant une séquence de 64 pas de temps, chacun avec un vecteur caractéristique de taille 768. Le modèle commence par une couche LSTM de 64 unités, suivie de deux autres couches LSTM de 128 et 192 unités respectivement. Ces couches LSTM sont configurées pour renvoyer des séquences, ce qui signifie que la sortie de chaque couche LSTM est introduite dans la suivante de la séquence.Après les couches LSTM, la sortie est aplatie en un vecteur 1D et passée à travers trois couches denses. La première couche dense a 1024 unités avec une activation ReLU et un terme de régularisation de la décroissance du poids L2 de 0,001. Une couche de décrochage avec un taux de 0,2 est appliquée après la première couche dense. La deuxième couche dense a 512 unités avec activation ReLU et les mêmes paramètres de régularisation et d'abandon. La troisième couche dense a 256 unités avec activation ReLU et les mêmes paramètres de régularisation et d'abandon.

Enfin, le modèle se termine par une couche dense ayant 8272 unités et une fonction d'activation linéaire. Cette couche produit la sortie finale du modèle. Le nombre total de paramètres pouvant être entraînés dans le modèle est de 3 538 256.

Le modèle est compilé à l'aide de l'optimiseur Adam [28] avec un taux d'apprentissage de 0,001. L'erreur quadratique moyenne (MSE) [27] est choisie comme fonction de perte et la précision est utilisée comme métrique pour l'évaluation pendant la formation.

Le modèle est entraîné à l'aide de la fonction d'ajustement, avec les tableaux d'entrée spécifiés pour les données d'entraînement et les données de validation. La taille du lot est définie sur 32 et le modèle est entraîné pendant 100 époques avec un brassage des données d'entraînement à chaque époque.

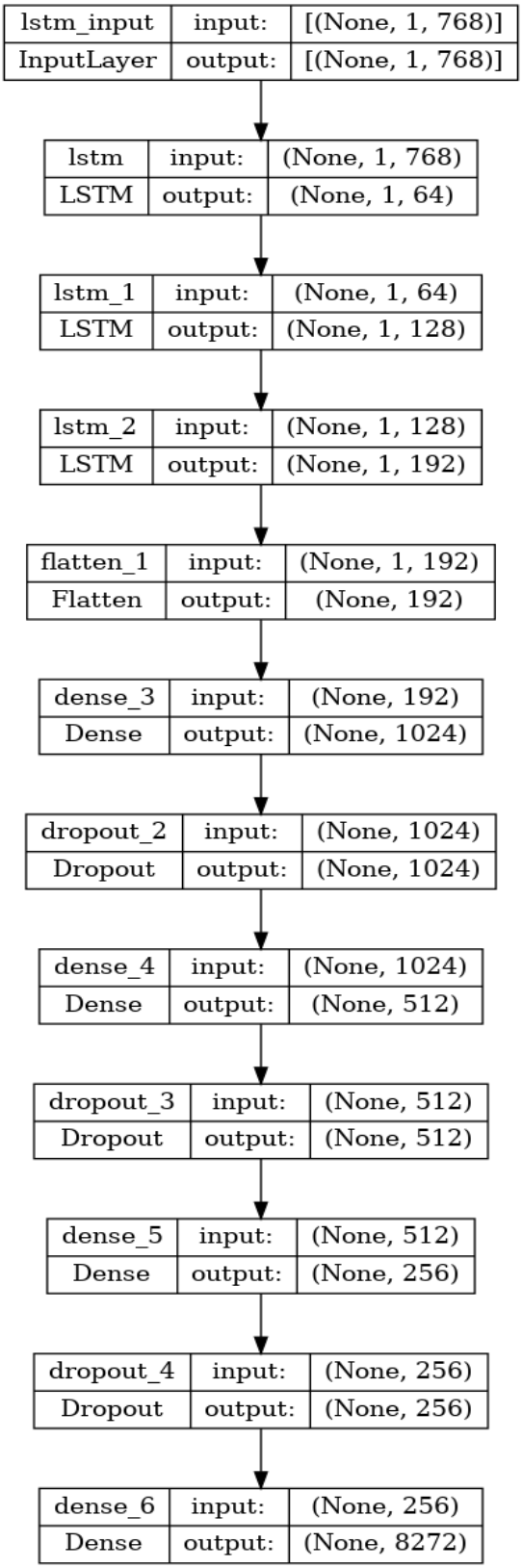


Figure 13: schéma de l'architecture RNN de la méthode 2

4. la fonction de perte:

Notre système modèle est formé en utilisant une approche d'optimisation basée sur le critère de classement [7]. L'objectif est de s'assurer que les clips audio et leurs sous-titres correspondants sont plus similaires les uns aux autres dans l'espace d'intégration par rapport aux paires de sous-titres audio incompatibles. Pour ce faire, un lot de N paires de sous-titres audio $\{(x_n, y_n)\}$ est considéré, où yn représente le sous-titre associé à un clip audio xn. Dans ce lot, des clips d'imposteur ($\hat{x}_n$) et des sous-titres d'imposteur ($\hat{y}_n$) sont sélectionnés au hasard pour chaque paire de sous-titres audio (x_n, y_n). La perte de triplet basée sur l'échantillonnage couramment utilisée [8, 9] est ensuite calculée pour capturer la relation entre les incorporations audio et de sous-titres. La fonction de perte, qui intègre cette perte de triplet, ainsi que le score de pertinence des sous-titres audio S, est optimisée pour former le système modèle. la perte de triplet est calculée avec:

$$loss = \frac{1}{N}\sum_{n=1}^{N}[\max(0, S(x_n, \hat{y}_n) - S(x_n, y_n) + 1) + \max(0, S(\hat{x}_n, y_n) - S(x_n, y_n) + 1)],$$

5. Conclusion:

Ce chapitre a présenté l'approche proposée pour la recherche de sons avec légende d'écriture humaine. L'objectif principal était d'expliquer chaque partie de l'architecture utilisée. Le chapitre a discuté de divers composants et techniques impliqués dans la réalisation de cet objectif. Celles-ci comprenaient les caractéristiques du texte, le stockage du son dans la base de données, l'utilisation de modèles appropriés tels que CNN et RNN et la prise en compte de l'importance de la fonction de perte. En examinant chacun de ces aspects, nous avons jeté des bases solides pour les chapitres suivants afin d'explorer plus avant la mise en œuvre et l'évaluation du système.

Chapitre 3: l'accomplissement

1. Introduction:

Ce chapitre donne un aperçu complet de la configuration expérimentale utilisée dans notre enquête sur la récupération audio à l'aide de sous-titres écrits par l'homme. Dans le chapitre précédent, nous avons présenté notre approche de cette tâche. Ici, nous approfondissons les outils utilisés dans le développement de notre système de récupération audio, qui englobe une gamme variée de technologies. De plus, nous fournissons une description détaillée de notre ensemble de données, y compris les sources et les caractéristiques de l'audio et des sous-titres utilisés dans nos expériences. En outre, nous décrivons la procédure que nous avons suivie pour analyser et discuter des résultats de nos expériences, qui impliquaient d'évaluer les performances de notre système à l'aide de diverses mesures telles que la précision et le rappel. Dans l'ensemble, ce chapitre sert de guide complet pour le développement et l'évaluation de notre système de récupération audio, et met en lumière l'efficacité de l'utilisation de sous-titres écrits par l'homme pour faire correspondre le contenu audio pertinent.

2. Outils utilisés:

La recherche en apprentissage en profondeur nécessite souvent de grands ensembles de données et des opérations de calcul intensives, soulignant l'importance du calcul parallèle pour accélérer le processus de formation. Bien que les GPU soient un choix courant à cette fin, leur coût élevé d'acquisition et de maintenance peut poser des risques tels que la dépréciation de l'équipement et une utilisation excessive. Pour répondre à ces préoccupations, nous avons adopté des alternatives rentables, telles que COLAB et Kaggle. COLAB, une plate-forme basée sur un navigateur, et Kaggle, une plate-forme de science des données renommée, nous ont permis d'effectuer des calculs gourmands en ressources et d'exécuter efficacement le code Python[13].

Utilisation utilisée: COLAB et Kaggle: nous avons utilisé la puissance de COLAB, une plate-forme basée sur un navigateur, et de Kaggle, une plate-forme de science des données populaires, pour effectuer des calculs gourmands en ressources et effectuer efficacement le code Python. [13] Ces alternatives économiques aux GPU nous permettent d'accélérer le processus de formation tout en gérant efficacement les coûts.De plus, nous avons utilisé Google Drive Storage pour stocker et gérer en toute sécurité nos ensembles de données, nos points de contrôle de modèles et nos résultats expérimentaux. Cela a facilité l'accès et le partage des données au sein de notre équipe de recherche tout en fournissant une solution de sauvegarde et de synchronisation fiable.

Développement: Jupyter, Conda et PyCharm [14]: nous avons utilisé Jupyter comme environnement de développement interactif pour l'expérimentation et le prototypage. Conda nous a aidé à gérer les packages et à créer des environnements reproductibles, assurant la cohérence de nos recherches. De plus, PyCharm [14] a fourni un environnement de développement intégré (IDE) robuste pour le codage, le débogage et la gestion de projet, amélioré ainsi notre workflow de développement.



Bibliothèque: Librosa [16], SBERT, BERT, Tkinter[15], Keras, TensorFlow: pour activer des fonctionnalités spécifiques dans nos recherches, nous avons utilisé une gamme de bibliothèques. Librosa [16] a fourni des capacités d'analyse audio complètes, tandis que SBERT (Sentence-BERT) a proposé des modèles de pointe pour les incorporations de phrases. BERT, un modèle basé sur un transformateur largement utilisé, enrichi nos tâches de représentation du langage. En conclusion de nos efforts, Tkinter [15]a joué un rôle central dans le développement d'interfaces utilisateur graphiques (GUI) pour nos applications interactives. Enfin, nous avons utilisé les capacités combinées de Keras et TensorFlow, deux cadres d'apprentissage en profondeur de premier plan, pour mettre en œuvre et expérimenter efficacement divers modèles d'apprentissage en profondeur..




3. Description de l'ensemble de données:

L'ensemble de données Clotho v2 [6] offre une immense valeur pour l'analyse et la récupération audio. Il se compose de 6974 échantillons audio accompagnés de 34 870 sous-titres, chaque échantillon ayant cinq légendes. Les extraits audio durent de 15 à 30 secondes et les sous-titres comptent de 8 à 20 mots. Cet ensemble de données est une ressource précieuse pour faire progresser et évaluer les algorithmes de récupération audio.

L'ensemble de données est collecté à partir de la plateforme Freesound [32] et les sous-titres sont obtenus à l'aide d'Amazon Mechanical Turk. Il est divisé en trois divisions: Développement, Validation et Évaluation. L'ensemble de données comprend des métadonnées stockées dans des fichiers CSV, fournissant des informations telles que le nom du fichier, les mots-clés, les URL et les détails du téléchargeur/utilisateur.

La gestion de la taille et des ressources du jeu de données est essentielle en raison du grand nombre de sous-titres et d'échantillons audio. Cependant, la structure bien organisée de l'ensemble de données Clotho v2 [6] facilite l'accès et l'analyse, ce qui en fait une ressource précieuse pour les chercheurs et les développeurs dans le domaine de l'analyse et de la récupération audio.

4. Métriques d'évaluation:

- Précision et Rappel d'un Classifieur Binaire/non-binaire : Pour les problèmes de classification binaire (où il n'y a que deux classes), la précision et le recal peuvent être calculés à l'aide des formules suivantes :

  Précision = TP / (TP + FP)

  Rappel = TP / (TP + FN)

  Pour les classificateurs non binaires, la précision et le rappel peuvent être calculés

individuellement pour chaque classe à l'aide de la même formule.

- Précision et rappel au seuil k: la précision et le rappel au seuil k sont des mesures d'évaluation qui prennent en compte les k meilleures prédictions faites par un modèle et calculent le rapport des vrais positifs (TP) à la somme des vrais positifs et des faux positifs (FP) dans Calcul de la précision et des vrais positifs (TP) à la somme des vrais positifs et des faux négatifs (FN) dans le calcul du rappel. Il est utile lorsque nous voulons évaluer les performances du modèle à un seuil ou à une coupure spécifique.

- Précision moyenne moyenne (MAP): est une métrique d'évaluation couramment utilisée dans les systèmes de recherche d'informations et de recommandation. Cette métrique fournit une évaluation complète des performances d'un modèle à différents niveaux de rappel. Elle est également particulièrement utile lorsqu'il s'agit de résultats de récupération classés. Il calcule la précision moyenne sur différents niveaux de rappel, pour ce faire, nous calculons la précision pour chaque niveau de rappel, puis nous calculons la précision moyenne sur les niveaux de rappel

- La métrique mAP a été largement utilisée pour évaluer les performances des algorithmes de récupération intermodaux [3]

- mAP@K (avec K=10): également connu sous le nom de "Mean Average Precision at 10", est une variante de MAP qui se concentre sur la précision et le rappel à un seuil spécifique de 10. Il mesure la précision moyenne des 10 meilleures prédictions faites par un modèle.

5. Évaluation et résultat:

Méthode 1:

- **Pour l'architecture du sBert méthode 1:**

  Les valeurs de la fonction de perte d'entraînement (trained for 300 epochs with triplet loss function):

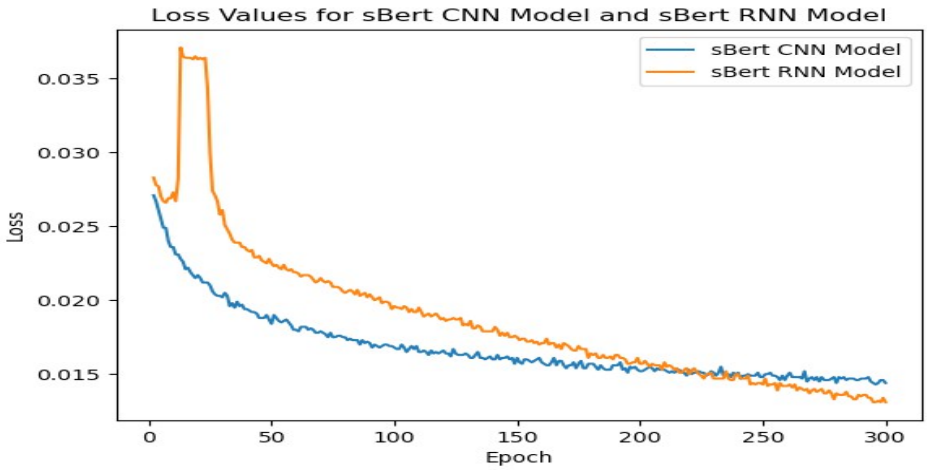


Figure 14: les valeurs de perte pour le modèle sBert CNN/RNN méthode 1

Nous avons calculé les 10 audio les plus proches pour chaque valeur de notre ensemble de données d'évaluation à l'aide du produit scalaire, le résultat de la partie évaluation:

Model 1		Model 2	
Métrique	Valeur	Métrique	Valeur
R1	0.075	R1	0.148
R5	0.179	R5	0.221
R10	0.251	R10	0.258
mAP10	0.072	mAP10	0.076

Tableue 7 - résultat de l'architecture sBert CNN/RNN méthode 1

- **Pour l'architecture du Bert méthode 1:**

  Les valeurs de la fonction de perte d'entraînement (trained for 100 epochs with triplet loss function):

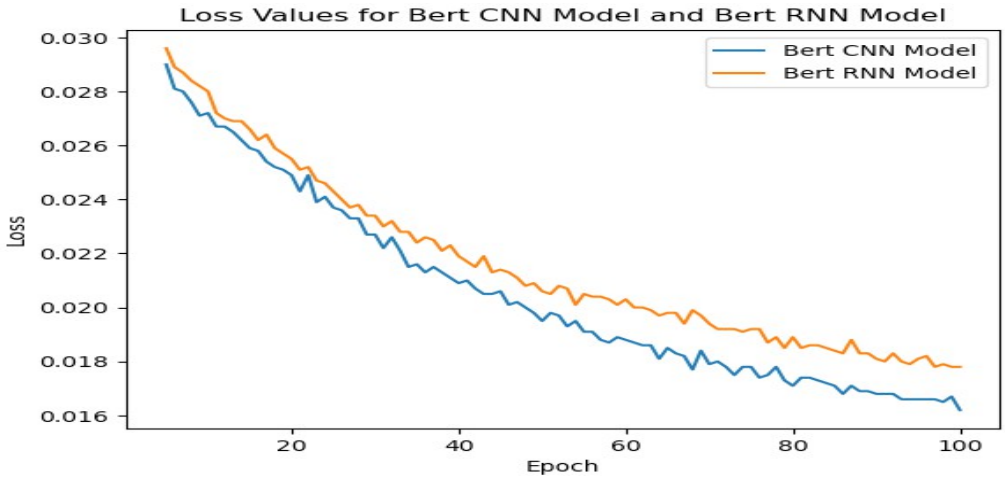


Figure 15: les valeurs de perte pour le modèle Bert CNN et le modèle Bert RNN méthode 1

Nous avons calculé les 10 audio les plus proches pour chaque valeur de notre ensemble de données d'évaluation à l'aide du produit scalaire, le résultat de la partie évaluation:

Model 3		Model 4	
Métrique	Valeur	Métrique	Valeur
R1	0.122	R1	0.309
R5	0.194	R5	0.348
R10	0.249	R10	0.367
mAP10	0.072	mAP10	0.107

Tableue 8 - résultat de l'architecture Bert CNN/RNN méthode 1

Méthode 2:

- **Pour l'architecture sBert méthode 2:**

Les valeurs de la fonction de perte d'entraînement (trained for 100 epochs with Mean Square Error (MSE) function [27]):

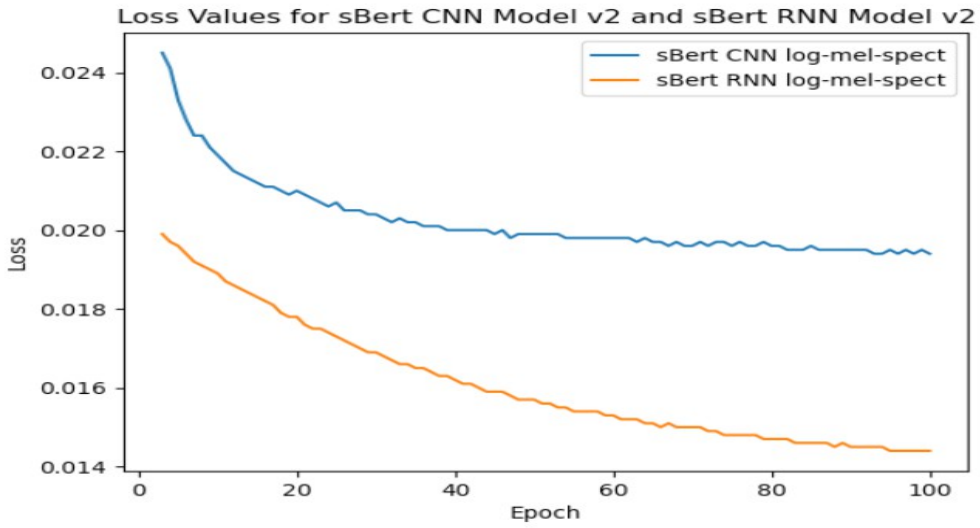


Figure 16: les valeurs de perte pour le modèle sBert CNN et le modèle sBert RNN méthode 2

Nous avons calculé les 10 audio les plus proches pour chaque valeur de notre ensemble de données d'évaluation en utilisant la similarité cosinus, le résultat de la partie évaluation:

Model 5		Model 6	
Métrique	Valeur	Métrique	Valeur
R1	0.027	R1	0.034
R5	0.075	R5	0.200
R10	0.127	R10	0.391
mAP10	0.037	mAP10	0.114

Tableue 9 - résultat de l'architecture sBert CNN/RNN méthode 2

- **Pour l'architecture du système de base 2022 et 2023:**

Le résultat de la partie d'évaluation pour les 10 audio les plus proches en utilisant le produit scalaire:

Word2Vec, CRNN, log mel spectrogram (2022)		sBert, CNN14, log mel spectrogram (2023)	
Métrique	Valeur	Métrique	Valeur
R1	0.03	R1	0.130
R5	0.11	R5	0.343
R10	0.19	R10	0.480
mAP10	0.07	mAP10	0.222

Tableue 10 - résultat de l'architecture du système de base 2022 et 2023

6. Discussion sur les résultats:

Dans cette section, nous présentons les résultats obtenus à partir des différentes architectures utilisées dans notre étude. Les mesures de performance, y compris R1, R5, R10 et mAP10, ont été évaluées pour différentes combinaisons d'architectures sBert et Bert avec Des modèles CNN et RNN à l'aide de spectrogrammes log mel.

model	R1	R5	R20	MAP10
Model 1	0.075	0.179	0.251	0.072
Model 2	0.148	0.221	0.258	0.076
Model 3	0.122	0.194	0.249	0.072
Model 4	0.309	0.348	0.367	0.107
Model 5	0.027	0.075	0.127	0.037
Model 6	0.034	0.200	0.391	0.114
Baseline 1 (2022)	0.03	0.11	0.19	0.07
Baseline 2 (2023)	0.130	0.343	0.480	0.222

Tableue 11 - comparaison de tous les modèles avec toutes les métriques d'évaluation

- sBert contre Bert: En comparant les architectures sBert et Bert, nous avons observé que les modèles Bert surpassaient systématiquement les modèles sBert sur plusieurs métriques. Les modèles Bert ont atteint des valeurs plus élevées pour R1, R5, R10 et mAP10 par rapport aux modèles sBert. Cela indique que l'architecture Bert, avec ses incorporations contextualisées, était plus efficace pour capturer les modèles audio et obtenir de meilleures performances de reconnaissance, ce qui signifie que sBert était également difficile pour le système d'apprendre de.

- CNN vs RNN: nous avons évalué les performances des modèles CNN et RNN dans chaque architecture. Pour les architectures sBert et Bert, les modèles RNN ont généralement surpassé les modèles CNN pour les métriques évaluées. Les modèles RNN ont atteint des valeurs plus élevées pour R1, R5, R10 et mAP10 par rapport aux modèles CNN. Cela suggère que la nature séquentielle des RNN, qui peuvent capturer les dépendances temporelles dans les données audio, était bénéfique pour les tâches de reconnaissance de formes audio.

- la méthode 1 et la méthode 2 de l'architecture: un constat notable se dégage : les performances de la méthode 1(Encoder-Encoder) surpassent celles de la méthode 2. Par conséquent, on en déduit que l'approche consistant à extraire des caractéristiques du spectrogramme log mel, plutôt que de le prédire directement, donne résultats supérieurs. Ce résultat souligne l'importance de l'extraction de fonctionnalités dans

l'amélioration des performances globales de l'architecture.

- la meilleure et la pire solution: le meilleur modèle est le modèle 4 et le pire modèle est le modèle 5

Compte tenu de ces résultats, l'architecture Bert avec un modèle RNN a présenté les meilleures performances globales dans nos expériences. Il a obtenu une plus grande précision de reconnaissance et de meilleurs résultats de récupération par rapport aux autres combinaisons.

7. Conclusion:

Dans ce chapitre, nous avons présenté notre approche de la récupération audio avec des sous-titres écrits par des humains. Nous avons d'abord présenté les outils et les langages de programmation utilisés, puis décrit le jeu de données utilisé pour nos expérimentations. Nous avons évalué notre approche à l'aide de divers paramètres et présenté les résultats et l'analyse. Notre approche a obtenu des résultats prometteurs, qui indiquent le potentiel de l'utilisation de sous-titres écrits par l'homme pour la récupération audio.

Conclusion:

Dans notre approche proposée pour la récupération audio à l'aide de sous-titres écrits par l'homme, nous avons fourni un aperçu complet des techniques utilisées pour le traitement audio et texte, et nous avons mis en évidence les travaux connexes dans ce domaine. Notre approche visait à tirer parti de la puissance des techniques avancées de traitement de texte telles que BERT et sBERT, ainsi qu'à stocker des données audio dans la base de données Clotho [6] pour obtenir de meilleurs résultats de récupération. De plus, nous avons utilisé diverses architectures de réseaux de neurones, notamment CNN, RNN, pour développer un modèle capable d'extraire des fonctionnalités à partir d'entrées textuelles et audio.

Nous avons mené plusieurs expériences pour évaluer l'efficacité de notre approche, en utilisant quatre architectures différentes qui combinaient l'utilisation de sBERT ou BERT avec CNN ou RNN pour obtenir des caractéristiques de spectrogramme log mel. Malgré certains défis dus aux limitations matérielles, nous avons obtenu des résultats prometteurs qui ont démontré le potentiel de l'utilisation de sous-titres écrits par l'homme pour améliorer les performances de récupération audio.

Nous avons également fourni une description détaillée des outils que nous avons utilisés, y compris l'ensemble de données Clotho [6] et les mesures d'évaluation. En outre, nous avons discuté des orientations futures de ce travail, telles que l'exploration de différentes architectures de réseaux neuronaux, l'intégration de fonctionnalités supplémentaires et la résolution des limitations matérielles auxquelles nous étions confrontés.

Dans l'ensemble, notre approche proposée pour la récupération audio à l'aide de sous-titres écrits par l'homme a le potentiel d'améliorer les performances des systèmes de récupération audio existants. Nous croyons que notre travail peut inspirer d'autres recherches dans ce domaine et conduire au développement de systèmes de récupération audio plus efficaces et efficients à l'avenir.

Travaux futurs:

Sur la base des résultats de notre approche proposée, plusieurs domaines de travaux futurs peuvent être explorés pour améliorer encore l'efficacité et l'efficience de la récupération audio à l'aide de sous-titres écrits par l'homme. Voici quelques directions potentielles pour de futures recherches :

Exploration de différentes architectures de réseaux de neurones : bien que nous ayons utilisé CNN et RNN dans notre approche proposée, de nombreuses autres architectures pourraient être explorées, telles que GNN [24], l'auto-encodeur ou des modèles pré-entraînés tels que BERT ou GPT [23]. Ces architectures pourraient être en mesure de mieux capturer les relations entre les données audio et textuelles et d'améliorer les performances.

Sous-titrage audio automatisé en temps réel : notre approche proposée reposait sur le traitement hors ligne des données audio et textuelles, ce qui peut ne pas convenir aux applications en temps réel. Les recherches futures pourraient donner la priorité à l'avancement de modèles conçus pour offrir un sous-titrage audio automatisé instantané. Ce développement s'avérerait inestimable dans diverses applications, y compris les scénarios de diffusion en direct ou de téléconférence.

Amélioration de la compatibilité matérielle : pour relever les défis posés par les limitations matérielles identifiées au cours de nos expériences, il serait avantageux de se concentrer sur le raffinement du modèle afin de minimiser la dépendance à des configurations matérielles spécifiques. Cette optimisation renforcerait l'évolutivité et améliorerait les performances globales.

Extension à de nouvelles plates-formes : afin d'élargir la portée et la convivialité de notre approche proposée, il serait utile d'étudier les possibilités d'intégrer le modèle dans des plates-formes émergentes telles que les applications mobiles ou les services Web. Cette expansion permettrait une plus grande accessibilité et garantirait que les avantages du modèle puissent être exploités par une base d'utilisateurs plus large.

Un autre domaine passionnant de travaux futurs est l'intégration des PANN [20] (réseaux de neurones audio pré-entraînés à grande échelle) pour la reconnaissance des formes audio. Les PANN tirent parti de l'apprentissage par transfert, où des modèles pré-entraînés capturent des fonctionnalités audio précieuses à partir d'ensembles de données à grande échelle. En affinant ces modèles sur des tâches spécifiques, les chercheurs peuvent capitaliser sur leurs connaissances audio générales, réduisant ainsi le besoin de données étiquetées étendues. L'exploration de nouvelles architectures et stratégies de formation dans le cadre PANN est prometteuse pour améliorer leur capacité à capturer des modèles audio complexes et à améliorer leurs performances dans diverses applications.

LES RÉFÉRENCES:

[1] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic Annotation and Retrieval of Music and Sound Effects" IEEE trans. audio. speech. Lang. Processing, vol. 16, no. 2, PP.467-476, 2008, doi: 10.1109/TASL.2007.913750

[2] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. InProceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014 Oct (pp. 1532-1543).

[3] P. Kaur, H. S. Pannu, and A. K. Malhi, "Comparative analysis on crossmodal information retrieval: A review," Comput. Sci. Rev., p. 100336, 2021.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.

[5] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in Proc. NACCL, 2019.

[6] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in Proc. ICASSP, 2020.

[7] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah, "Signature ¨ verification using a "siamese" time delay neural network," in Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS), 1993, pp. 737–744.

[8] H. Xie, O. Ras¨ anen, K. Drossos, and T. Virtanen, "Unsupervised ¨ audio-caption aligning learns correspondences between individual sound events and textual phrases," in Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP), 2022, pp. 8867–8871.

[9] D. Harwath, A. Torralba, and J. R. Glass, "Unsupervised learning of spoken language with visual context," in Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS), 2016, pp. 1866–1874

[10] Agarap AF. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375. 2018 Mar 22.

[11] A. Paszke, S. Gross, F. Massa, et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, et al., Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-animperative-style-high-performance-deep-learning-library.pdf

[12] H. Xie, S. Lipping, and T. Virtanen, "DCASE 2022 Challenge Task 6B: Language-Based Audio Retrieval," 2022. [Online]. Available: https://arxiv.org/abs/2206.06108

[13] Python, "Python," https://www.python.org/downloads/release/python-394/, Apr. 04, 2021.

[14] Fonctionnalités de PyCharm, "https://www.jetbrains.com/help/pycharm/quick-start-guide.html."

[15] Lundh F. An introduction to tkinter. URL: www. pythonware. com/library/tkinter/introduction/index. htm. 1999.

[16] Mcfee et al, "http://conference.scipy.org/proceedings/scipy2015/pdfs/brian_mcfee.pdf," Librosa - audio processing Python library, 2015.

[17] A. Biswal, «Recurrent Neural Network (RNN) Tutorial: Types, Examples, LSTM and More, » 21

02 2022. [En ligne]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn. [Accès le 07 03 2022].

[18] «https://openclassrooms.com/fr,» [En ligne]. Available: https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donneesvisuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn.

[19] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084. 2019 Aug 27.

[20] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," IEEE/ACM Trans. Audio Speech Lang. Process., pp. 2880–2894, 2020.

[21] Alsentzer E, Murphy JR, Boag W, Weng WH, Jin D, Naumann T, McDermott M. Publicly available clinical BERT embeddings. arXiv preprint arXiv:1904.03323. 2019 Apr 6.

[22] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. *Signature verification using a "siamese" time delay neural network.* In Proc. 6th Int. Conf. Neural Inf. Process. Syst. (NIPS), 737–744. 1993.

[23] Ethayarajh K. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. arXiv preprint arXiv:1909.00512. 2019 Sep 2.

[24] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE transactions on neural networks. 2008 Dec 9;20(1):61-80.

[25] Reynolds DA. Gaussian mixture models. Encyclopedia of biometrics. 2009 Jul 2;741(659-663).

[26] Rahutomo F, Kitasuka T, Aritsugi M. Semantic cosine similarity. InThe 7th international student conference on advanced science and technology ICAST 2012 Oct (Vol. 4, No. 1, p. 1).

[27] Wan AT, Ohtani K. Minimum mean-squared error estimation in linear regression with an inequality constraint. Journal of Statistical Planning and Inference. 2000 Apr 15;86(1):157-73.

[28] Zhang Z. Improved adam optimizer for deep neural networks. In2018 IEEE/ACM 26th international symposium on quality of service (IWQoS) 2018 Jun 4 (pp. 1-2). Ieee.

[29] Chowdhary K, Chowdhary KR. Natural language processing. Fundamentals of artificial intelligence. 2020:603-49.

[30] K. Sohn, "Improved deep metric learning with multi-class npair loss objective," in Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, et al., Eds., vol. 29. Curran Associates, Inc., 2016.

[31] E. Fonseca, X. Favory, J. Pons, et al., "FSD50K: an open dataset of human-labeled sound events," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 829–852, 2022, publisher: IEEE.

[32] F. Font, G. Roma, and X. Serra, "Freesound Technical Demo," in Proceedings of the 21st ACM International Conference on Multimedia, ser. MM '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 411–412, event-place: Barcelona, Spain. [

[33] T. Wolf, L. Debut, V. Sanh, et al., "Transformers: State-of-the-Art Natural Language Processing," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online: Association for Computational Linguistics, Oct. 2020,

pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6

[34] Y. Liu, M. Ott, N. Goyal, et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[35] Mei, Xinhao, et al. "Language-based audio retrieval with pre-trained models." *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge, Tech. Rep* (2022).

[36] Primus, Paul, and Gerhard Widmer. "Improving Natural-Language-based Audio Retrieval with Transfer Learning and Audio & Text Augmentations." *arXiv preprint arXiv:2208.11460* (2022).

[37] Weck, Benno, et al. "Matching Text and Audio Embeddings: Exploring Transfer-Learning Strategies for Language-Based Audio Retrieval." *arXiv preprint arXiv:2210.02833* (2022).