Université Paris III - Sorbonne Nouvelle

Mémoire de Master 2 Traitement Automatique des Langues
Parcours Recherche et Développement

Présenté par Natalia Kalashnikova

Classification automatique avec SVM et CNN de la parole expressive

Sous la direction de Nicolas Audibert

Année universitaire 2019 - 2020

Remerciements

Je tiens à remercier le directeur de mémoire, Monsieur Nicolas Audibert, pour sa patience, ses explications et ses corrections.

Je voudrais également remercier Audrey et Alexandre qui ont pris le temps de relire et de corriger ce mémoire.

TABLE DES MATIÈRES

Résumé

La classification automatique des émotions est largement utilisée dans plusieurs domaines, comme la relation client, la santé, les banques, etc. Malgré l'intérêt croissant de cette problématique, il n'existe pas de standard de combinaison de descripteurs d'apprentissage optimal, ni d'algorithme de référence pour sa performance. Ainsi, ce mémoire présente la comparaison de performances des SVM et des CNN. Le SVM est appliqué aux différents types de données, y compris les données audio. Le CNN à deux dimensions est utilisé pour la classification des images avec une très haute performance. La performance des CNN à une dimension est peu étudiée pour la classification des émotions à l'oral. Ainsi, nous avons appliqué l'approche du CNN à une dimension de Li et al. (2019) qui a été utilisée pour la classification des émotions. Le corpus RECOLA annoté en continu en polarité a été segmenté en extraits de 2 à 2,5 secondes représentant les polarités neutre, positive et négative. Après la segmentation, le corpus d'apprentissage contient peu de données annotées en polarité négative. Le SVM et le CNN à une dimension ont été entraînés sur des statistiques calculées pour toute la durée des extraits mais aussi sur des valeurs de MFCC extraites à partir des fenêtres glissantes de 25 ms. Les images des MFCC ont été générées à partir des valeurs et ont été utilisées pour l'entraînement du CNN à deux dimensions. Le MFCC, qui est l'un des descripteurs cepstraux, est souvent utilisé pour la reconnaissance des émotions à l'oral, grâce aux critères statistiques et leurs bonnes performances dans les tâches de classification. Nous avons découvert que le CNN est mieux adapté pour la classification de la polarité dans le cas du corpus déséquilibré et des extraits à durée différente. Le CNN à une dimension entraîné sur les valeurs de MFCC a montré de meilleurs résultats que le CNN à deux dimensions développé sur les images. L'approche de Li et al. (2019) a donc prouvé son efficacité. Le meilleur résultat a été obtenu par le CNN à une dimension entraîné sur les statistiques de MFCC.

Introduction

L'oral est le moyen le plus rapide et naturel d'interagir entre les humains. Ceci a motivé les chercheurs à projeter ce moyen sur la communication homme-machine pour la rendre plus efficace et rapide. La communication humaine est basée sur le contenu linguistique et émotionnel. La parole porte l'information linguistique et paralinguistique associée aux émotions. Ainsi, la reproduction de la communication homme-homme à homme-machine nécessite un haut niveau de reconnaissance de la parole et de compréhension des émotions. Le problème de la reconnaissance des émotions est traité dans le domaine de l'analyse des sentiments, qui vise à définir l'état émotionnel du locuteur à partir du langage. Ceci reste toujours un des défis principaux du TAL (El Ayadi et al. 2011).

Le classifieur automatique des émotions nécessite un modèle théorique des émotions. Les travaux de recherche (e.g. Xiao et al., 2005; Origlia et al., 2010) se concentrent souvent sur la détection des émotions de base (Izard, 1971; Ekman, 1972; Plutchik, 1984). Les émotions peuvent également être décrites sous l'approche multidimensionnelle (Gunes & Pantic, 2010; Gunes et al., 2011). Dans cette approche, les sentiments sont représentés sous trois dimensions : la polarité ou la valence (est-ce que c'est positif ou négatif ?), l'*arousal* ou l'activation (est-ce que c'est fort ou faible ?) et la dominance (est-ce que c'est dominant ou dominé ?).

Les études de Lugger et Yang (2007), Kim et al. (2010) et Jeon et al. (2011) montrent que certaines émotions comme la joie et la colère, ou la tristesse et le neutre ont des propriétés acoustiques similaires. Du point de vue de l'approche multidimensionnelle, les émotions dans ces paires ont le même niveau de l'*arousal*. A son tour, l'approche unidimensionnelle propose de décrire les émotions par le critère de la polarité (Thomas & Znaniecki, 1918). Cette approche est appliquée à la création des modèles de classification dans le cadre de ce travail, afin d'éviter le biais des patrons acoustiques similaires entre des émotions du même niveau d'activation. Ainsi, on compare la performance de deux algorithmes d'apprentissage, qui sont le SVM (*Support Vector Machine* ou Machine à vecteurs de support) et le CNN (*Convolutional neural network* ou Réseau neuronal convolutif), appliqués à la classification automatique de la polarité.

Le SVM est appliqué aux différents types de données, y compris les données audio. Le CNN à deux dimensions est utilisé pour la classification des images avec une très haute performance. La performance des CNN à une dimension est peu étudiée pour la classification des émotions à l'oral. Ainsi, nous comparons l'approche du CNN à une dimension de Li et al. (2019) avec l'application traditionnelle du CNN à deux dimension entraînés sur les images.

Ces algorithmes sont développés sur des valeurs et des statistiques de MFCC. Ce sont des coefficients cepstraux qui sont calculés à partir du spectre du signal acoustique auquel la transformée en cosinus discrète a été appliquée.

Les valeurs de MFCC sont calculés à partir des fenêtres glissantes de 25 ms avec une durée de chevauchement de 10 ms et sont normalisées. Ces valeurs sont utilisées comme descripteurs pour l'entraînement du SVM et du CNN à une dimension. De plus, les images sont générées à la base de ces valeurs pour l'entraînement du CNN à deux dimensions. La moyenne, l'écart-type, la dérivée première moyenne et la dérivée seconde moyenne sont calculés pour chaque extrait. Ces statistiques sont utilisées comme des descripteurs pour l'entraînement du SVM et du CNN à une dimension.

Les statistiques de MFCC illustrent les valeurs globales des extraits, les valeurs de MFCC permettent d'analyser l'évolution des coefficients MFCC au cours du temps. L'utilisation traditionnelle du CNN à deux dimensions rend possible l'évaluation du CNN à une dimension appliquée au traitement de l'oral.

De nombreuses lectures ont été effectuées afin de compléter le contexte théorique sur ce sujet. Le développement des modèles a été précédé par un pré-traitement des données. Cette étape comprenait la segmentation des données, l'extraction des MFCC, la génération des images et l'adaptation des valeurs des descripteurs au format de l'input des modèles.

Le Chapitre 1 de ce mémoire aborde le sujet des théories des émotions et clarifie les définitions des notions des émotions, des attitudes et de la polarité. La première partie du Chapitre 2 définit les domaines d'application et les problématiques de l'analyse des sentiments. Ensuite, les différents descripteurs d'apprentissage sont décrits. Les algorithmes utilisés dans ce travail sont présentés à la fin de ce chapitre. Et, enfin, le Chapitre 3 présente dans un premier temps les guides d'annotation pour le corpus utilisé dans ce mémoire. Il détaille également les étapes de pré-traitement des données et de développement des modèles. Ce chapitre est conclu par une discussion sur les résultats.

1. Émotions, attitudes, polarité

La notion d'émotions est le sujet d'étude de plusieurs disciplines scientifiques, telles que la psychologie, la biologie, la linguistique, etc. Chaque discipline apporte une partie des connaissances à la notion d'émotions. Néanmoins, il n'existe pas une définition commune, comme il n'existe pas une théorie unique d'études des émotions même au sein d'une seule discipline. De plus, plusieurs termes, comme les émotions et les affects sociaux ou attitudes, sont employés pour désigner les sentiments humains. Dans ce chapitre, on analyse les différentes approches des théories des émotions et des attitudes ainsi que la notion de polarité (valence).

1.1. Émotions

Le domaine de la psychologie distingue quatre théories des émotions : évolutionniste, physiologique, cognitive et la théorie du constructivisme social.

La théorie évolutionniste, fondée par Darwin (1872), considère que les émotions ont un rôle adaptatif et social et sont universelles. Cette approche a été ensuite développée par Izard (1971), Ekman (1972) et Plutchik (1984) qui ont menés leurs recherches autour des émotions de base (primaires ou fondamentales), qui sont universelles à travers les cultures. Les six émotions communes parmi les chercheurs sont la colère, le dégoût, la peur, la tristesse, la joie et la surprise. Les émotions complexes sont vues comme une combinaison des émotions primaires (Lu, 2015). Cette approche considère que les émotions réalisent une fonction adaptative pour la survie d'un individu. Avec le développement de cette théorie, les chercheurs (Wallon, 1934, 1970 ; Malrieu, 1952 ; Dumas, 1948) ont déclaré que les émotions exercent une autre fonction, qui est la régulation du comportement en fonction des demandes sociales.

La perspective physiologique (James, 1902; Lange, 1895) définit les émotions comme le résultat des changements physiologiques. Par exemple, mes jambes tremblent donc j'ai peur. Ce point de vue s'oppose à l'approche de Cannon (1927) et Bard (1928), qui considèrent que le traitement d'un stimuli externe par le système nerveux déclenche une émotion. Par exemple, je vois un danger, j'ai peur, donc mes jambes tremblent. Ainsi, l'activation physiologique seule, sans activation du système nerveux, ne mène pas au ressenti des

émotions. La réaction physiologique est donc vue plutôt comme la conséquence des émotions et pas leur origine. Les deux branches de cette théorie sont liées par la manifestation physiologique des émotions.

L'activation physiologique figure également dans la théorie cognitive de Schachter et Singer (1962). Cette théorie considère que les émotions peuvent être décrites par deux critères : l'activation physiologique qui détermine l'intensité du sentiment, et la cognition qui interprète la situation et identifie la nature du sentiment. La notion de cognition est centrale dans cette théorie. Les émotions sont différenciées par l'évaluation d'une situation par un participant (Arnold, 1960).

L'approche du constructivisme social étudie les émotions du point de vue social (Averill, 1980). Les émotions sont considérées comme un ensemble de réponses sociales, c'est-à-dire des normes et des attentes au sein d'un contexte social. Ainsi, pour cette approche, les émotions ne sont pas universelles à travers différentes cultures. De même, au sein de la même culture, la même situation peut provoquer des émotions différentes dans des groupes sociaux différents. Cette réaction est expliquée par la différence de classe sociale, d'âge, de préférences politiques, etc. (Gergen, 1985).

On peut constater que la notion d'émotion n'a pas de définition théorique commune. Néanmoins, Fontaine et al. (2007) et Scherer (2001) ont développé une théorie qui a établi un consensus plus large parmi les chercheurs de différentes disciplines qui étudient les émotions. Cette théorie définit les émotions comme les moments de changements de plusieurs composants, tels que l'activation neurophysiologique, l'expression motrice et le sentiment subjectif en réponse aux événements internes ou externes. Les émotions différentes peuvent être décrites selon 4 dimensions : *evaluation-pleasantness* (dans cette dimension les émotions plaisantes sont opposées aux émotion déplaisantes), *potency-control* (cette dimension est caractérisée par le sentiment du contrôle de la situation et de soi-même), *activation-arousal* (cette dimension décrit la force de la réaction), *unpredictability* (cette dimension prédit la surprise de l'émotion).

1.2. Attitudes

La notion de l'attitude a été d'abord utilisée pour désigner les réactions d'une personne envers une situation dans le domaine de la psychologie sociale (Thomas & Znaniecki, 1918).

Dans l'approche d'Allport (1935), l'attitude est créée à la base de l'état mental et neurologique qui se fonde sur l'expérience d'une personne et lui dicte donc un comportement envers une situation.

La réaction d'une personne peut être mesurée selon une dimension positif/négatif ou favorable/défavorable. Il s'agit donc de la perspective unidimensionnelle (Thomas & Znaniecki, 1918). Cette perspective considère que l'attitude représente l'évaluation d'une situation. Cette approche s'inscrit donc dans le cadre de la théorie cognitive des émotions.

Le modèle multidimensionnel analyse l'attitude selon 3 critères : le sentiment, la cognition et le comportement (Rosenberg & Hovland, 1960). Le critère sentimental est lié à l'affect, le critère cognitif est associé "aux croyances ou opinions évoquées par l'objet d'attitude" (Lu, 2015, p.15), et le critère comportemental réfère aux actions (Vallerand & Lafrenaye, 2006). Dans la même perspective, Osgood (1966) a proposé le modèle d'évaluation des attitudes selon les trois critères suivants : l'évaluation (positive/négative), l'activation (forte/faible) et le contrôle (volontaire/involontaire).

Un modèle hybride a été proposé par Zanna et Rempel (1988). Dans cette approche, l'attitude est d'abord définie par le modèle unidimensionnel, et ensuite par le modèle multidimensionnel. Par conséquent, l'attitude est caractérisée selon ces quatre critères : le jugement sur la situation, l'affect, les opinions et les actions.

1.3. Polarité

Dans la littérature scientifique, les termes "valence" et "polarité" sont utilisés pour désigner le même concept. Dans ce mémoire, ces deux termes sont utilisés comme des synonymes et sont donc équivalents.

Le corpus utilisé pour la détection automatique de la parole expressive a été annoté en polarité et en *arousal* - degré de réaction aux stimuli (pour les détails d'annotation, voir Chapitre 3.1). Le modèle caractérisant l'affect par les deux critères primaires, valence et arousal, a été développé par Russel (1980).

Les émotions peuvent être vues comme des sous-catégories de la polarité (Ortony et al., 1988). Par conséquent, les émotions peuvent être soit positives, soit négatives. Certaines émotions peuvent être associées à une valence, c'est le cas de la joie qui est toujours associée à la polarité positive, ou de la colère qui est liée à la polarité négative. L'association d'autres

émotions à une seule valence est plus problématique et dépend souvent du contexte. Par exemple, les chercheurs distinguent la surprise positive et la surprise négative (Noordewier & Breugelmans, 2013). Ainsi, la distinction entre ces deux types de surprise a conduit Ortony, Clore et Collins (1988) à ne pas considérer la surprise comme une émotion.

Comme Russel (1980), Williams et Stevens (1981) ont réalisé des études physiologiques pour analyser les émotions de deux dimensions : polarité et arousal. Les chercheurs ont trouvé que le système nerveux sympathique est activé par la joie, la colère et la peur, qui sont des émotions d'une forte activation. Ces émotions provoquent une augmentation du rythme cardiaque et de la tension, l'augmentation du volume respiratoire, un assèchement de la bouche et des tremblements musculaires occasionnels. Dans la même perspective Cahn (1990) a fait la conclusion que la parole est réalisée dans les hautes fréquences et devient forte et rapide. Le système nerveux parasympathique est activé par l'activation faible (par exemple, la tristesse) et induit des réactions physiologiques inverses : la diminution de la tension et du rythme cardiaque, et la production d'une grande quantité de salive. La parole est donc lente et dans les basses fréquences. De plus, pour le chercheur les paramètres prosodiques, comme la fréquence fondamentale, la durée et la qualité de la voix, sont en corrélation avec l'expression des émotions. Néanmoins, les émotions ne peuvent pas se distinguer uniquement avec l'*arousal* (activation). Ainsi, la colère et la joie sont des émotions d'activation forte, mais elles transmettent des affects différents qui sont associés avec la polarité.

Scherer (1986) a étudié le lien entre la valence et les conséquences sur la production de la parole. Dans ce travail, la polarité est vue sous la dimension d'agréable / désagréable (*pleasant / unpleasant*). Ainsi, la polarité agréable se manifeste par l'expansion du pharynx. De plus, le conduit vocal raccourcit car les coins des lèvres montent. Au niveau du spectre il y a plus d'énergie dans les basses fréquences, la bande du premier formant est un peu plus large et le contour mélodique de la F0 est tombant. La polarité désagréable provoque la constriction du pharynx. Le conduit vocal est également plus court, mais parce que les coins des lèvres sont baissés. Sur le spectre, il y a plus d'énergie dans les hautes fréquences, la bande du premier formant est étroite, son contour mélodique est montant, mais les contours mélodiques du deuxième et du troisième formants sont tombants.

Conclusion au Chapitre 1

La plupart des théories sur les émotions considèrent que les émotions sont universelles à travers les cultures, qu'elles sont involontaires et servent pour la survie physiologique et sociale d'un individu. Les attitudes, à leur tour, sont définies comme l'intention individuelle, qui est influencé par les normes sociales et la situation donnée.

La notion de polarité est réciproquement liée avec des émotions et des attitudes. Ainsi, les émotions peuvent être partiellement décrites par la dimension de la polarité, comme elles peuvent être caractérisées en sous-catégories de la polarité. De même, la polarité peut être vue comme une projection unidimensionnelle de l'attitude.

Dans ce mémoire, la polarité est considérée comme la dimension la plus informative pour décrire des émotions et des attitudes. Ainsi, pour réaliser la tâche de la classification automatique, on utilise donc le corpus annoté en polarité et pas en étiquettes des émotions.

2. Analyse des sentiments

Généralement, l'analyse des sentiments définit l'attitude de quelqu'un envers un sujet exprimé par les moyens langagiers. Dans ce cas, l'attitude signifie un état émotionnel : un jugement (positif ou négatif), une émotion (joie, colère, peur, etc.), une humeur, etc. Une des tâches de l'analyse des sentiments est la détection automatique de la polarité. Autrement dit, cette tâche vise à classifier un morceau du langage écrit ou parlé selon trois catégories : neutre, positif et négatif. De cette façon, l'analyse des sentiments peut être définie comme une tâche de détection automatique des sentiments dans le langage (Mohammad, 2016). Dans ce chapitre, des exemples de l'application, des difficultés ainsi que les descripteurs d'apprentissage de l'analyse des sentiments dans le texte et dans la parole sont présentés.

2.1. Analyse des sentiments dans le texte

L'analyse peut être réalisée au niveau de la détection des sentiments du lecteur ou d'autres participants de la communication, mais principalement il se focalise sur l'analyse des sentiments de l'auteur.

Les sentiments peuvent être détectés à différents niveaux : mots, propositions, SMS, tweets, avis, documents, etc. Le sens même de certains mots peut contenir une connotation précise. Par exemple : "bien", "mal". D'autres mots ont une très forte association à une valence positive ou négative. Par exemple : "mort" est fortement lié à une polarité négative, et "fête" à une polarité positive. Les mots qui n'ont pas d'association à une valence sont considérés comme neutres. Ces associations sont utilisées dans les vocabulaires (lexiques) de la valence, où une polarité (positive ou négative) est associée aux mots. Au niveau des propositions, une étiquette de polarité est attribuée aux propositions entières.

Il existe également la détection des sentiments envers un concept dans le monde. Par exemple, un avis sur un produit ou un service, des opinions envers le monde (politique, droits, etc.). Cette tâche est liée à l'opération de détection de la subjectivité qui vise à regrouper les propositions dans deux classes : celles qui contiennent des opinions et des attitudes, et sont donc considérées comme subjectives, et celles qui constatent des faits, et sont donc considérées comme objectives (Mohammad, 2016).

L'analyse des sentiments dans le texte est largement utilisée dans les domaines différents comme la santé et la biomédecine, la politique, l'éducation, l'art (Mohammad, 2016).

La détection automatique des sentiments dans le texte est problématique pour plusieurs raisons, comme (Mohammad, 2016) :

- La valeur émotionnelle d'une unité langagière (proposition ou morceau de texte) n'est pas égale à la somme de ses composants (mots). On peut imaginer un avis sur un restaurant laissé sur un site (la deuxième ligne indique la valence de chaque mot de la proposition) :

"Le	restaurant	est	pas	mal"
0	0	0	-	-

  La somme de la polarité des mots de la proposition est négative, alors que la valence générale de toute la proposition est positive. De plus, les mêmes mots utilisés dans des contextes différents peuvent transmettre des émotions différentes. Par exemple, le mot "terrible", dans l'expression 'ce groupe est terrible", transmet l'idée d'une personne extraordinaire. La proposition peut donc être jugée comme ayant une polarité positive. Tandis que dans l'expression "c'est un accident terrible", le mot "terrible" évoque l'idée d'un événement catastrophique, qui a donc une valence négative. Il est aussi fréquent que les émotions ne soient pas exprimées de façon explicite.
- Le contenu des réseaux sociaux est souvent utilisé (Kanakaraj & Guddeti, 2013; Phand & Phand, 2017) pour l'analyse des sentiments, car il est riche en termes non-standard, comme les fautes d'orthographe, les mots transcrits phonétiquement mais qui ne correspondent pas aux normes orthographiques, les hashtags, les emojis, les abréviations, etc. En absence de prosodie, les utilisateurs expriment leurs émotions à travers ces termes.
- Les algorithmes d'apprentissage automatique nécessitent une grande quantité de données annotées pour l'entraînement d'un modèle et pour son évaluation. De plus, la perception des émotions peut être compliquée même pour les humains. Elle dépend de l'humeur de la personne qui s'exprime mais aussi de la personne réceptrice, de leur contexte culturel, expérience personnelle, etc. Ces facteurs influencent la communication quotidienne, mais aussi l'annotation des données. Ainsi, l'annotation nécessite des instructions claires, un entraînement des annotateurs, des données de bonne qualité, etc.

Dans l'analyse de la polarité à l'écrit, le texte est souvent représenté sous la forme de vecteurs de traits. Par exemple, le modèle contient 4 traits binaires qui sont présentés sous la forme de 0 et de 1. Le texte sera donc représenté au format <0, 0, 1, 1>, qui caractérise le texte par ces 4 traits. Les descripteurs d'apprentissage utilisés par les systèmes de classification sont variés : les n-grams de mots, les vocabulaires (lexiques) d'association de mot à une polarité, les POS (parties du discours), les ponctuations, les relations syntaxiques, etc.

La classification de la polarité a été considérablement améliorée (Socher et al., 2013, Le & Mikolov, 2014) par la méthode de plongement des mots. Cette technique représente des mots sous la forme d'un vecteur de nombres réels. Ainsi, les mots ayant des vecteurs proches apparaissent dans des contextes similaires. Contrairement à la méthode de vecteurs des traits choisis pour l'apprentissage, les traits de cette technique ne peuvent pas être interprétés directement.

2.2. Reconnaissance des émotions à l'oral

L'analyse des sentiments à l'oral est généralement réalisée à partir du signal de parole, sans analyser le contenu linguistique. Les techniques appliquées dans l'analyse des sentiments pour le discours parlé extraient l'information paralinguistique à partir du signal acoustique. Les descripteurs d'apprentissage sont fondés sur les éléments prosodiques, comme la fréquence fondamentale (F0), l'intensité et la durée, mais aussi sur les paramètres en lien avec le contenu spectral et avec la qualité de voix.

La reconnaissance de l'état émotionnel à l'oral est particulièrement utile dans les applications qui demandent l'interaction naturelle. Le but principal dans ces applications est d'adapter la réponse du système si l'agacement ou la frustration sont détectés (El Ayadi et al., 2011).

- Santé et biomédecine. De même qu'à l'écrit, la reconnaissance des émotions peut être utilisée comme un outil de diagnostic pour les médecins, afin de détecter la dépression et les risques de suicide (France et al., 2000).
- Défense. Dans les postes de pilotage, les systèmes de reconnaissance de la parole, qui sont entraînés sur la parole contenant du stress, montrent de meilleurs résultats que les systèmes entraînés sur la parole neutre (Hansen et Cairns, 1995).

- Centres d'appels. Le système de l'analyse de sentiments permet de détecter le moment "déclencheur" après lequel l'appelant perd le contrôle de ses émotions. Ceci pourrait aider le conseiller à réagir, et éventuellement à changer de technique pour prévenir cette perte de contrôle (Ma et al., 2006).
- Relation client. Pour mieux gérer le flux téléphonique, de plus en plus d'entreprises utilisent des assistants virtuels, qui sont capables d'adapter leur réponse et/ou de transférer le client vers un conseiller humain.
- Système embarqué dans les voitures. L'information sur l'état émotionnel du chauffeur permet au système de contrôler sa sécurité (Schuller et al., 2004).

La tâche de l'analyse des sentiments à l'oral est l'un des défis principaux dans le domaine du TAL pour plusieurs raisons (El Ayadi et al., 2011).

- Il n'existe pas de standard de combinaison des descripteurs d'apprentissage optimal ou même de descripteurs plus efficaces pour distinguer les émotions.
- L'hétérogénéité des propositions, locuteurs et styles de discours introduisent la variabilité acoustique, qui rend la tâche de la détection des émotions plus complexe.
- Un énoncé peut contenir plus d'une émotion. De plus, la notion d'émotion n'a pas de définition théorique commune. Les frontières entre les émotions ne sont donc pas définies.
- L'expression d'une émotion dépend du locuteur, de sa culture et de son contexte social.
- Quand une personne est dans un état émotionnel depuis longtemps (e.g. la tristesse), l'émotion détectée par le système n'est pas claire : celle à longue durée ou transitoire.

L'extraction des descripteurs représente une étape importante dans la construction des classifieurs et doit traiter 2 questions. La première est celle des frontières temporelles, car les descripteurs peuvent être extraits à partir de la durée de tout l'énoncé, et le signal acoustique peut être également divisé en intervalles, pour lesquelles les descripteurs d'apprentissage sont définis. La deuxième est celle des descripteurs d'apprentissage. Est-ce que le signal acoustique contient assez d'information pour distinguer des émotions différentes, ou il est nécessaire de le combiner avec d'autres sources comme les données linguistiques, faciales et contextuelles ?

Ces questions sont discutées dans les parties suivantes du chapitre. Tout d'abord, les différents types de descripteurs non-linguistiques sont décrits. Cette partie explique les

MFCC - le descripteurs cepstral utilisé dans le cadre de ce mémoire. Ensuite, les descripteurs linguistiques disponibles pour compléter les données non-linguistiques sont brièvement expliqués.

2.2.1. Descripteurs non-linguistiques

Le signal acoustique peut contenir plusieurs émotions, il est donc courant de diviser le signal en segments plus petits. Cette technique permet de désambiguïser l'information du signal acoustique (Rabiner & Schafer, 1978). Les paramètres prosodiques extraits des segments sont les descripteurs locaux. Les descripteurs globaux sont les statistiques calculées pour toute la durée de l'énoncé. Certaines études (Picard et al., 2001, Ververidis & Kotropoulos, 2003, Shami & Kamel, 2005, Hu et al., 2007) ont montré que les modèles entraînés sur les descripteurs globaux sont plus rapides et proposent une meilleure précision. Néanmoins, les descripteurs globaux sont efficaces uniquement pour la classification en émotions fortement activées ou faiblement activées (Nwe et al., 2003). Par exemple, la classification des extraits en joie et tristesse. Ils sont donc peu efficaces pour la classification du même niveau d'activation (le cas de la classification en joie et colère). De plus, l'information temporelle de la parole est perdue avec l'utilisation des descripteurs globaux.

Une autre méthode est l'extraction des descripteurs à partir des phones (Lee et al., 2004). L'étude de (Leinonen & Hiltunen, 1997) a montré les différences du spectre du même phonème en fonction des émotions exprimées. Cette méthode est rarement utilisée puisqu'elle nécessite la transcription phonétique ou l'alignement du signal acoustique.

Une approche plus facilement appliquée car on peut séparer automatiquement les parties voisées et non-voisées, propose l'extraction des descripteurs uniquement à partir des parties voisées de la parole (Rabiner & Schafer, 1978). Néanmoins, cette technique est peu utilisée, parce que les pauses silencieuses peuvent être les indices d'un état émotionnel ou d'une attitude.

Les descripteurs extraits du signal acoustique peuvent être catégorisés en 3 groupes : continus, qualitatifs et spectraux. Les descripteurs qualitatifs sont liés avec la qualité de voix et portent les étiquettes comme tendu (*tense*), dur (*harsh*), soufflé (*breathy*). Leur rôle n'est pas suffisamment étudié (Scherer, 1986; Gobl & Chasaide, 2003) pour les deux raisons suivants : 1) le manque du consensus entre les chercheurs sur les étiquettes de la qualité de

voix et donc leur lien avec les émotions ; 2) la complexité de la détection automatique des extraits (El Ayadi et al., 2011). Ainsi, dans cette partie du travail, on se concentre sur les descripteurs continus et spectraux.

Les descripteurs continus comprennent (El Ayadi et al., 2011) :

- Fréquence fondamentale (F0) - la fréquence la plus intense sur le spectre ;
- Energie - mesures de l'intensité - la puissance transportée par les ondes sonores ;
- Durée ;
- Formants - zones avec une augmentation d'énergie dans le spectre.

Les études (Cowie et al., 2001; Tao et al., 2006; Borchert & Dusterhoft, 2005) ont montré que les émotions peuvent être détectées grâce aux descripteurs prosodiques. Néanmoins, certaines émotions comme la colère, la joie, la peur et la surprise possèdent des caractéristiques de fréquence fondamentale similaires (Rabiner & Schafer, 1978; Cahn, 1990). Les différentes recherches indiquent les différents niveaux du débit de parole qui sont associés à la colère (Murray & Arnott, 1993 pour le débit de parole augmenté et Oster & Risberg, 1986 pour le débit de parole ralenti).

Comme les descripteurs continus, les descripteurs spectraux sont utilisés pour la représentation des segments du signal acoustique. L'information émotionnelle des segments est reflétée dans la distribution de l'énergie des fréquences (Nwe et al., 2003). Ainsi, les énoncés réalisés avec la joie ont plus d'énergie dans les hautes fréquences, tandis que les énoncés avec la tristesse ont peu d'énergie dans les mêmes fréquences (Banse & Scherer, 1996). Pour une meilleure exploitation, la distribution spectrale est passée par des filtres et les descripteurs spectraux sont extraits à partir des résultats de ces filtres.

Autres que les descripteurs spectraux, il existe les descripteurs cepstraux, dont les plus utilisés sont les MFCC. Pour calculer le vecteur MFCC, le signal acoustique est d'abord passé par le filtre pour renforcer l'énergie dans les hautes fréquences. Ensuite, le signal est divisé en petits segments de 30 ms environ qui se chevauchent pour capturer la continuité du signal. Les étapes suivantes comprennent la transformation de Fourier rapide et l'application des filtres de l'échelle de Mel. La valeur du logarithme de chaque spectre de Mel est prise et la transformée en cosinus discrète est appliquée pour la création du vecteur MFCC (Tursunov et al., 2019).

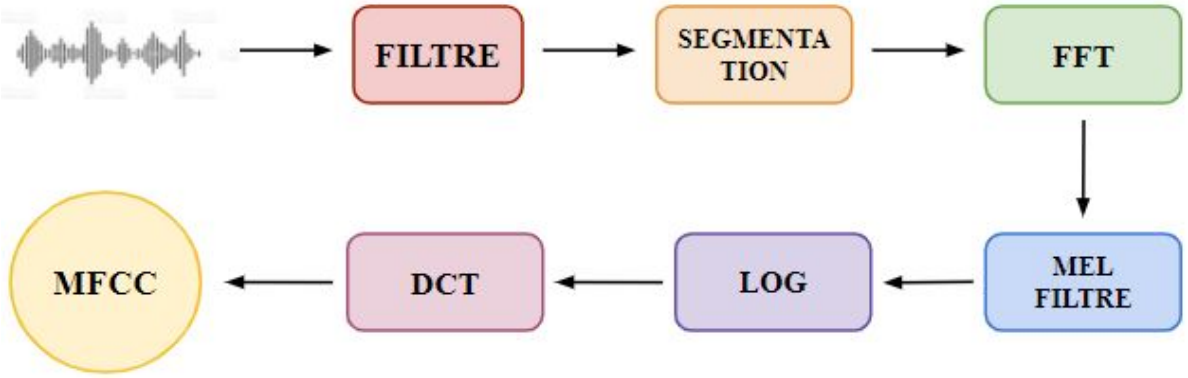

Image 1. Schéma de l'extraction des MFCC

Les chercheurs (Tursunov et al., 2019; Shrawankar & Thakare, 2013) déclarent que les modèles entraînés sur les MFCC montrent de hautes performances sur la classification de la polarité et de l'arousal (activation). Les MFCC sont largement utilisés dans le domaine de l'analyse des sentiments puisqu'ils sont calculés à l'aide de l'échelle de Mel, qui est relative à l'échelle de l'oreille humaine. Dans ce mémoire, le modèle de la classification des sentiments est entraîné sur les valeurs de MFCC.

Les descripteurs d'apprentissage pour l'audio, comme pour le texte, sont souvent représentés sous la forme de vecteur. Il existe deux classifications de descripteurs d'audio. En fonction de leurs structures temporelles, les descripteurs sont catégorisés comme segmental (de courte durée) et suprasegmental (de longue durée). Les descripteurs segmentaux font référence aux phones et sont calculés pour les extraits de 25-50 msec, ce qui permet d'analyser leur évolution temporelle. A leur tour, les descripteurs suprasegmentaux font référence aux variations prosodiques et sont calculés pour l'énoncé entier.

Dans l'étude récapitulative sur les descripteurs, El Ayadi et al. (2011) a fait la conclusion que les descripteurs cepstraux, comme les MFCC, sont le meilleur moyen de représenter la parole pour la classification des sentiments.

Une autre information non-linguistique est exprimée par les vocalisations non-linguistiques, telles que rires, pleurs, soupirs, etc. Néanmoins, le lien entre les vocalisations et les états émotionnels concrets n'as pas encore été suffisamment étudié. Par exemple, le rire peut être la suite d'un amusement qui a été provoqué par une blague, et il peut également être la conséquence du sarcasme qui est lié au mépris (Russel et al., 2003).

2.2.2. Descripteurs linguistiques

Le contenu linguistique joue un rôle important dans l'expression des émotions (El Ayadi et al., 2011). Les descripteurs linguistiques utilisés pour la classification des sentiments à l'oral sont les mêmes qui sont utilisés pour l'analyse des sentiments dans le texte. Ces descripteurs sont extraits du texte de l'enregistrement si le corpus audio représente la parole lue, ou de la transcription de l'audio si le corpus représente la parole spontanée.

Ainsi, Batliner et al. (2003) ont montré que les répétitions et les reformulations sont les marques de la hausse de l'irritation et de la colère. La même recherche a permis de découvrir que les noms et les adjectifs sont plus saillants que les verbes pour la classification des émotions.

Lee et Narayanan (2005) ont étudié comment la présence d'un mot peut influencer la catégorie émotionnelle de l'énoncé. De cette façon, les chercheurs ont réuni en un dictionnaire des mots associés à une classe négative ou non-négative.

Le travail de Forbes-Riley et Litman (2004) a montré que l'information supplémentaire sur la nature de l'émotion peut être tirée du contexte. Les sources comme le genre, une pause longue avant la réponse, un changement soudain du débit et de l'intensité sont donc des indices pour la classification automatique des émotions à l'oral.

Une autre technique utilisée pour améliorer l'analyse des sentiments à l'oral avec l'information linguistique est l'analyse sémantique. Ijima et al. (2009) ont proposé d'utiliser les mots-clés. De même, Wu et Liang (2011) ont introduit la méthode de l'étiquetage sémantique du texte ou de la transcription de l'audio. Néanmoins, ces systèmes manquent de ressources sémantiques et syntaxiques.

L'utilisation d'autres sources comme les vidéos, les transcriptions et l'information psychologique sur les participants peut améliorer la performance d'un classifieur des émotions (Cohen, 1984; Grosz & Sidner, 1986).

Le corpus RECOLA utilisé dans ce mémoire pour la construction du classifieur automatique ne contient pas de transcription des enregistrements. De plus, les systèmes de la reconnaissance de la parole nécessitent une étape de post-traitement importante. Nous avons donc décidé de se concentrer sur le développement d'un modèle uniquement à base de descripteurs extraits du signal audio.

2.3. Choix du corpus

Les données utilisées dans le domaine de la reconnaissance des émotions à l'oral peuvent être dépendantes ou pas du locuteur (speaker dependent/independent). Les corpus dépendant du locuteur visent à reconnaître les émotions des participants des enregistrements dans des contextes différents. Le système entraîné sur ce type de corpus obtient une bonne performance, mais uniquement sur les voix des personnes enregistrées. Les corpus indépendants du locuteur ont pour but de reconnaître les caractéristiques générales des expressions émotionnelles supposées communes à l'ensemble des personnes.

La performance d'un classifieur est influencée par la nature des données utilisées pour son entraînement. Les enregistrements de la parole spontanée dans des situations réelles améliorent sa performance, car ces systèmes sont conçus pour être appliqués à la parole spontanée (El Ayadi et al., 2011 ; Batliner et al., 2000).

Néanmoins, la plupart des corpus annotés en émotions sont enregistrés dans les laboratoires. Il existe deux approches principales pour provoquer l'expression des émotions chez les participants. Dans la 1ère approche, on demande à des professionnels (acteurs) de réagir comme s'ils étaient dans un état émotionnel particulier. Une autre approche utilise la technique du magicien d'Oz, qui aide les participants à entrer dans les états émotionnels (Batliner et al., 2000). Une méthode plus récente propose d'utiliser les jeux vidéo, où le déroulement du jeu induit les émotions (Johnstone et al., 2005)

Une autre question pour le choix du corpus est la distribution des émotions. Ainsi, une méthode consiste à utiliser une distribution équilibrée des émotions dans le corpus afin de réaliser une évaluation exacte d'un classifieur (Burkhardt et al., 2005). Une autre méthode propose de garder les mêmes proportions des émotions dans le corpus que dans la vie réelle (Morrison et al., 2007, You et al., 1997). De cette façon, les énoncés émotionnellement neutres sont les plus fréquents dans le corpus.

2.4. Contexte de la classification automatique de la polarité à l'oral

L'étude de Goudbeek et Scherer (2010) a conclu que les émotions positives possédaient moins de variation de l'intensité, ainsi qu'une pente spectrale plus abrupte que les émotions

négatives. De plus, les descripteurs spectraux ainsi que le débit de la parole jouaient le rôle dominant pour la classification de la polarité. Les descripteurs cepstraux liés aux descripteurs spectraux ont montrés leur efficacité pour la détection de la valence dans l'étude d'Eyben et al. (2013).

Dans une autre étude, Eyben et al. (2016) a proposé un ensemble minimal des descripteurs acoustiques pour la classification automatique. Cet ensemble comprend des descripteurs relatifs à la fréquence fondamentale, à l'énergie, au spectre et à la durée. La classification binaire a obtenu l'exactitude de 95,3% pour l'arousal et de 78,1% pour la polarité.

Dans la recherche de Tursunov et al. (2019) le timbre est considéré comme un ensemble des caractéristiques d'un son qui permet de le distinguer avec des autres ayant la même intensité et la fréquence fondamentale. Cette étude a montré que les descripteurs de timbre sont efficaces pour la classification de la polarité. Ces descripteurs sont surtout performants pour améliorer la classification des émotions positives.

L'étude de Zhang et al. (2019) a proposé une nouvelle architecture du CNN pour la classification de la polarité, de l'*arousal* et de la dominance afin d'éviter l'*over-fitting*. Dans la structure proposée une couche d'attention a été rajoutée pour chaque tâche avant la couche de prédictions. Ce modèle a montré de meilleurs performances par rapport à la baseline.

2.5. Algorithmes d'apprentissage

Les algorithmes utilisés pour l'apprentissage automatique sont variés et ont chacun leurs avantages et défauts. Il n'existe donc pas un consensus sur l'algorithme le plus adapté à la classification des émotions dans la parole (El Ayadi et al., 2011). Dans ce mémoire, on compare la performance d'une méthode souvent utilisée pour la détection de la valence, qui est le SVM, et une méthode utilisée plus rarement dans le cadre de l'analyse des sentiments, qui est le CNN. Dans cette partie du travail, on va donc présenter ces deux algorithmes.

2.5.1. SVM

Le SVM (*Support Vector Machine* ou Machine à vecteurs de support) est un des algorithmes le plus utilisé pour la classification automatique, notamment pour la classification des émotions en vue de son optimisation globale. Il est surtout utilisé pour la classification binaire, mais peut également être appliqué dans la détection de plusieurs classes. L'idée de

base sur laquelle l'algorithme est fondé est la création d'un hyperplan entre les classes, placé afin de maximiser la valeur des marges. Dans le cas particulier d'un espace à deux dimensions, cet hyperplan consiste en une simple ligne. Ici, les marges sont les distances entre les représentants les plus proches de la classe et la ligne. Les représentants les plus proches sont appelés les vecteurs de support (Tursunov et al., 2019). Sur l'image ci-dessous, la ligne noire représente la ligne qui sépare les deux classes pour la classification, les lignes bleue et rouge marquent les limites de chaque classe, les points entourés sur ces lignes sont les vecteurs de support et la distance entre la ligne noire et les lignes rouge et bleue est la marge (Audiffren, 2017).

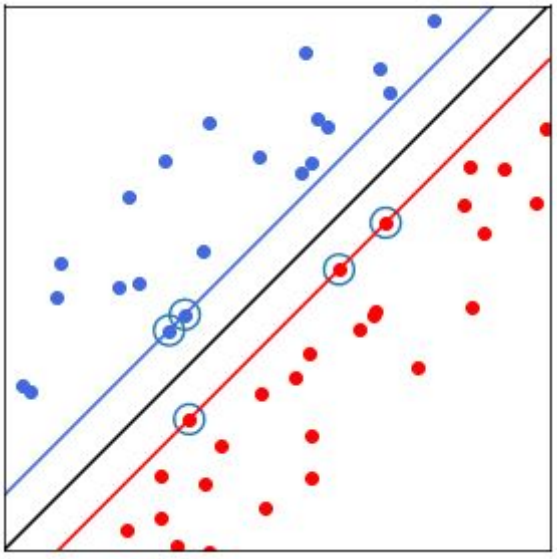

Image 2. Schéma de l'algorithme SVM appliqué aux données linéairement séparables

Néanmoins, cette représentation suppose que les données sont séparables de manière linéaire. Dans la plupart des cas, les données langagières ne sont pas séparables de manière linéaire. Par conséquent, l'algorithme utilise la fonction mathématique "noyau", qui permet de transformer les données dans un plus grand espace vectoriel (Audiffren, 2017).

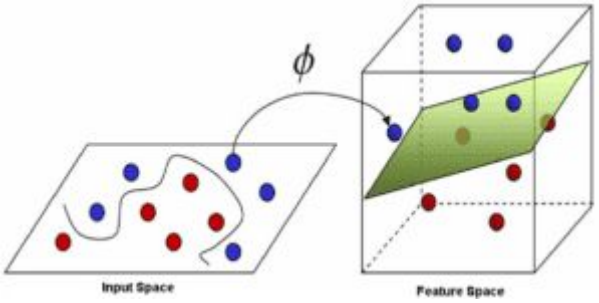

Image 3. Schéma de l'algorithme SVM dans un espace vectoriel large

Le SVM utilise également la technique de la maximisation de marge, qui rend le modèle plus robuste pour le bruit. Ainsi, le modèle SVM est plus généralisable et est utilisé pour différents types des données (dataanalyticspost, s.d.).

Le travail de Ke et al. (2018) a comparé la performance du SVM avec le réseau de neurones artificiel (ANN). Ces modèles ont été entraînés sur les descripteurs acoustiques (la F0, les MFCC, etc.) et statistiques (les valeurs maximales, minimales, moyennes, etc.). Dans cette étude le score de la reconnaissance des émotion est plus haut pour le modèle SVM. Dans l'article de Yu et al. (2011) sur le chinois le score de l'*accuracy* du SVM dépasse le score des modèles du LDC (*Linear discriminant classifiers*), du KNN (K plus proches voisins), RBFNN *(radial basis function neural network*). Seehapoch et Wongthanavasu (2013) ont utilisé le SVM pour la reconnaissance des émotions en allemand, japonais et thaï et ont obtenu respectivement le score de l'*accuracy* de 89,8%, 93,57% et 98%. L'utilisation des descripteurs cepstraux et spectraux pour le SVM développé pour le chinois et l'allemand a permis d'obtenir le score de 91,3% et 95,1% respectivement dans l'étude de Pan et al. (2012).

2.5.2. CNN

Les réseaux de neurones convolutifs sont le plus souvent utilisés pour la classification des images. Le CNN contient deux couches alternantes : de convolution et de pooling. Dans les couches de convolution, le système extrait des caractéristiques de haut niveau à l'aide de filtres. Les valeurs de filtres sont convoluées avec la matrice d'origine. Pour obtenir l'ensemble des caractéristiques complet (*feature map*) d'une couche de convolution, le filtre correspondant à cet ensemble (sur l'exemple, chaque ensemble de caractéristiques est représenté par les couleurs rouge, vert et orange) est glissé sur la matrice d'input. Ainsi, on obtient trois nouvelles matrices qui correspondent aux ensembles (*feature map*) obtenus. Pour la deuxième couche de convolution, la somme de convolutions pour chaque filtre par ensemble (*feature map*) est calculée (Wieser et al., 2018).

La couche de pooling utilise une opération de réduction d'échelle pour diminuer la taille de l'ensemble des caractéristiques (*feature map*) généré par la couche de convolution. Par cette opération, le système devient plus robuste pour la localisation spatiale des éléments dans l'image. La fonction la plus souvent utilisée pour diminuer la taille est la fonction de

max-pooling, qui prend uniquement les valeurs maximales de la fenêtre du filtre pour la couche suivante (Wieser et al., 2018).

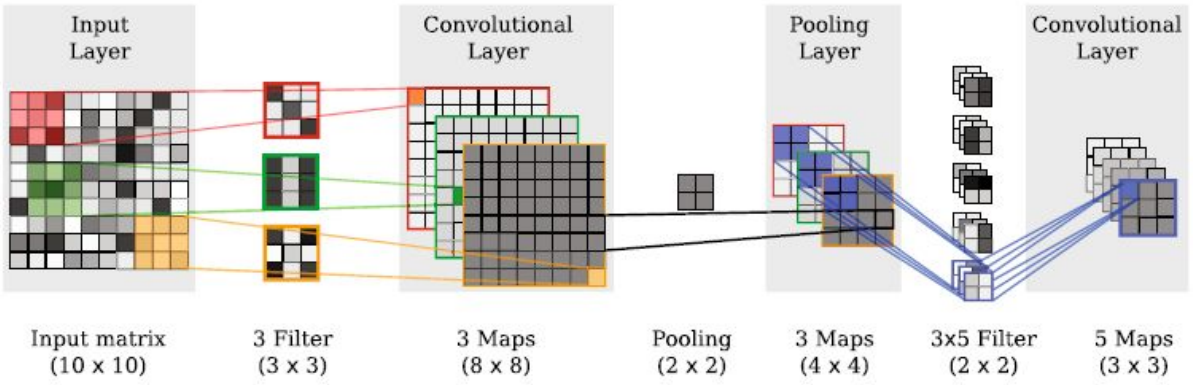

Image 4. Schéma de l'algorithme CNN (Wieser et al., 2018)

Après chaque combinaison des couches de convolution et de pooling, la fonction d'activation non-linéaire ReLU est activé. Les chercheurs supposent qu'elle est analogue à l'oreille humaine (Trigeorgis et al., 2016). Le travail de (Krizhevsky et al., 2012) a montré que la fonction ReLU augmente la convergence de la descente de gradient stochastique en comparaison avec d'autres fonctions d'activation. Dans les CNN, les neurones ont le même poids d'apprentissage sur la même couche, ce qui permet de diminuer le nombre de paramètres à apprendre (Wieser et al., 2018).

Dans le domaine du traitement automatique de la parole, les CNN prennent comme input les spectrogrammes ou la représentation des MFCC en images (Ferragne et al., 2019).

La reconnaissance automatique des émotions par le CNN a été étudiée dans le travail de Zhao et al. (2018). Dans cette méthode, le signal acoustique et les spectrogrammes ont été employés comme input au système hybride des réseaux de neurones convolutifs. Les résultats de cette étude montrent que l'utilisation des systèmes hybrides permet d'augmenter significativement la performance de la classification automatique des émotions. Dans le travail de Zheng et al. (2015) le score CNN entraîné sur les spectrogrammes a été meilleur que le celui du SVM.

Dans le cas de la classification des images, le système prend l'input à deux dimensions. Mais les données peuvent également être représentées dans un espace à une ou trois dimensions. La classification à une dimension se diffère de la classification à deux dimensions par l'application du filtre et le format des données pour l'input.

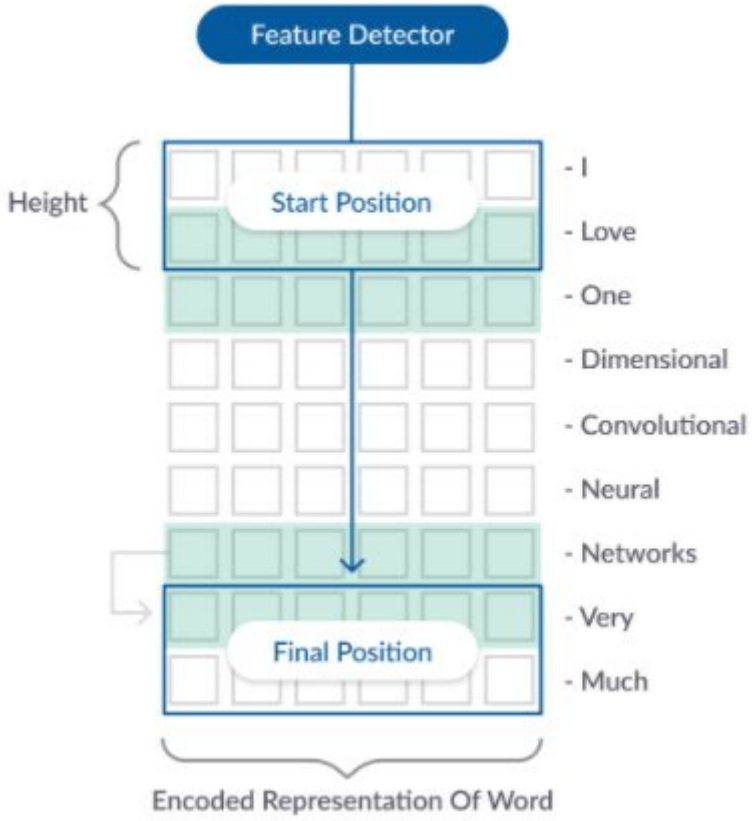

Image 5. Le schéma de CNN à une dimension ("Keras Conv1D: Working with 1D Convolutional Neural Networks in Keras", s.d.)

Dans l'exemple présenté la taille du filtre est 2, pour parcourir toutes les données le filtre doit donc passer 8 fois.

La performance des CNN à une dimension pour la reconnaissance des émotions a été testée dans le travail de Li et al. (2019). Le modèle a été entraîné sur les MFCC, les melspectrogrammes et les log-melspectrogrammes extraits avec la bibliothèque *librosa*. Le réseau neuronal contenait donc 6 couches de convolution à une dimension, 2 couches *dense* et 2 couches de *dropout* (la fonction des couches est expliquée dans le Chapitre 3.5).

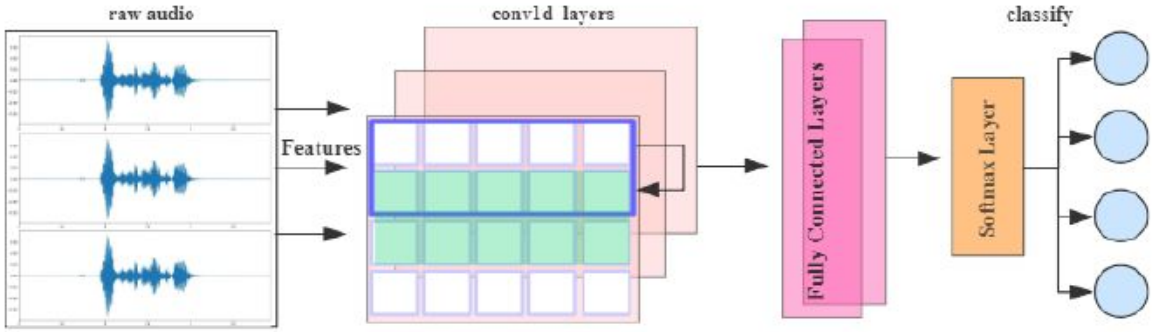

Image 6. Le schéma de CNN de Li et al. (2019)

Le modèle baseline a été entraînée uniquement sur les valeurs de MFCC et a obtenu l'accuracy de 63,55 pour le corpus IEMOCAP (Busso et al., 2008), 86,5 pour le corpus EMO-DB (Burkhardt et al., 2005) et 74,63 pour le corpus RAVDESS (Livingstone & Russo, 2018). L'utilisation des log-melspectrogrammes comme des descripteurs complémentaires a permis d'augmenter le score de 2-3% selon le corpus testé.

L'utilisation des CNN à une dimension est rare dans le domaine de la classification des émotions. Cependant, à travers le travail de Li et al. (2019) on peut remarquer que c'est une option intéressante et qui devrait être plus explorée. C'est pourquoi, dans ce mémoire on reprend cette idée pour l'appliquer à la classification de la polarité.

2.6. Métriques d'évaluation de la classification automatique

Dans le domaine du machine learning la performance des modèles sur la classification automatique est souvent mesurée selon les métriques suivantes : l'*accuracy* (nommée en français par l'exactitude ou la justesse), le rappel, la précision et la F-mesure.

L'accuracy indique la proportion des échantillons prédits correctement par le modèle par rapport au nombre total des échantillons à classer.

Accuracy = nombre de cas classés correctement / nombre total de cas à classer

Par exemple, un modèle doit classer 200 échantillons, dont 100 appartiennent à la classe neutre et 100 à la classe positive. Le système a correctement classé 50 échantillons de la classe positive et 90 échantillons de la classe neutre. L'accuracy est donc égale à (50 + 90) / 200 = 0,7.

Les autres métriques sont mesurées pour chaque classe de la classification.

La précision représente la proportion des échantillons classés correctement par rapport à tous les échantillons étiquetés dans cette classe.

Précision = nombre de cas d'une classe étiquetés correctement / nombre de cas étiquetés comme appartenant à cette classe

Le rappel mesure la proportion des cas étiquetés correctement par rapport à tous les échantillons de cette classe.

Rappel = nombre de cas d'une classe étiquetés correctement / nombre de cas de cette classe

Par exemple, le système a classé comme positif 60 échantillons. De ces 60 échantillons 50 sont réellement positifs et les autres 10 appartiennent à la classe neutre. Dans ce cas, la précision est égale à 50 / 60 = 0,83 et le rappel est égal à 50 / 100 = 0,5.

Le classifieur peut avoir un score élevé de précision s'il classe correctement très peu de documents. Par exemple, le système étiquette 10 échantillons comme positifs qui sont réellement positifs. La précision est donc de 1. De même, si le système classe tous les échantillons dans une seule classe, le score de rappel sera élevé pour cette classe. Il est donc important de tenir compte des deux métriques en même temps, mais aussi d'analyser la F-mesure, qui est la moyenne harmonique de rappel et de précision.

$$f\text{-}mesure = 2 * (précision * rappel) / (précision + rappel)$$

Cette métrique permet de prendre en compte à la fois le rappel et la précision.

Conclusion au Chapitre 2

Le domaine du traitement automatique des langues définit l'analyse des sentiments comme la détection automatique du contenu émotionnel dans le langage. A l'oral comme à l'écrit, la classification automatique des émotions partage les mêmes problèmes comme le manque des données annotées et disponibles pour la recherche, la présence de plusieurs émotions dans un énoncé / un texte, la dépendance des émotions du contexte culturel et social, etc. L'analyse des sentiments est appliquée dans plusieurs domaines, tels que la santé, les relations clients, la politique, les centres d'appels, etc.

La classification automatique des émotions à l'oral doit traiter deux questions lors du pré-traitement. Quelles sont les frontières temporelles pour l'extraction des descripteurs d'apprentissage ? Et, quels sont les descripteurs d'apprentissage ? Pour l'oral, il est possible d'utiliser deux types de descripteurs, les descripteurs linguistiques, qui sont également utilisés pour l'analyse des sentiments à l'écrit, et les descripteurs non-linguistiques. Le MFCC, qui est un des descripteurs cepstraux, est souvent utilisé pour la reconnaissance des émotions à l'oral, grâce aux critères statistiques et leurs bonnes performances dans les tâches de classification.

Plusieurs algorithmes d'apprentissage peuvent être utilisés pour l'entraînement des modèles de la reconnaissance des émotions. Nous avons choisi un modèle général qui est le SVM, un

algorithme souvent utilisé pour cette tâche. Un autre algorithme est le CNN, qui est appliqué plus rarement à la tâche de la classification des émotions.

3. Classification automatique de la polarité dans la parole

Le but de ce mémoire est de répondre à la question “quelle représentation des données et quel classifieur permet de mieux réaliser la classification automatique de la valence dans la parole”. Dans cet objectif, nous avons comparé la performance de deux méthodes, qui sont le réseau de neurones convolutifs (CNN) et la machine à vecteurs de support (SVM). Nous avons développé ces modèles sur deux types de données. Le SVM et le CNN à une dimension ont été entraînés sur les statistiques de MFCC (la moyenne, l’écart-type et la moyenne des dérivées première et seconde) calculées pour la durée entière de l’extrait. Les mêmes modèles ont été également développés sur les valeurs de MFCC extraites à partir des fenêtres glissantes de 25 ms. De plus, nous avons créé des images à partir des mêmes valeurs de MFCC pour l’entraînement du CNN à deux dimensions. La comparaison entre le CNN à une dimension et le CNN à deux dimensions permet d’analyser si le CNN à une dimension est performant sur les données de la parole, comme Li et al. (2019) l’ont déclaré.

Les descripteurs ont été extraits du corpus RECOLA qui est annoté en valence et en arousal. La suite de ce mémoire est organisée ainsi : la partie 3.1 détaille le corpus RECOLA, la partie 3.2 présente les étapes du pré-traitement des données, les parties 3.3. et 3.4 décrivent respectivement les modèles et les résultats de SVM et CNN. Ce chapitre est suivi par la conclusion de ce mémoire.

3.1. Corpus RECOLA

Malgré l'intérêt croissant des recherches sur les interactions sociales, les corpus de données spontanées et émotionnellement riches sont toujours manquants. Ce manque est dû à la complexité de la collecte de ce type de données, car l’expression émotionnelle des interactions sociales est de courte durée, relativement rare et dépendante des changements du contexte (Ringeval et al., 2013).

Pour la construction du corpus, 27 femmes et 19 hommes étudiants du département de psychologie de l’Université de Fribourg ont été enregistrés. Les 46 participants (l’âge moyen est de 22 ans, l’écart-type est de 3 ans) ont été regroupés en équipe de 2 personnes. Tous les participants sont francophones, parmi ceux-ci 33 ont le français pour langue maternelle, 8 - l’italien, 4 - l'allemand et 1 - le portugais. Le corpus est multimodal (audio, vidéo, données

physiologiques comme ECG (électrocardiogramme) et EDA (activité électrodermale)) mais dans ce mémoire, on traite uniquement des données audio.

Les participants ont été invités à résoudre la tâche de survie d'abord séparément et ensuite ensemble. Cette tâche a consisté à trouver le consensus dans le classement des 15 objets selon leur importance pour la survie de l'équipage dans le cas d'un écrasement d'avion dans un endroit hostile et déserté. La réalisation de ce type de tâche provoque une discussion intense dans le groupe et une réaction émotionnelle due aux enjeux liés à une bonne performance de l'exercice (la survie de l'équipage).

Les audios ont été enregistrés avec des microphones unidirectionnels sur le logiciel Audacity.

Pour l'annotation, les chercheurs ont gardé uniquement les 5 premières minutes d'enregistrements. Ce choix est guidé par plusieurs facteurs. Premièrement, les participants ont passé plus de temps sur la discussion des stratégies au début de l'exercice. Deuxièmement, les mêmes personnes ont annoté les enregistrements en polarité et en comportement social. La restriction de la quantité de données à annoter a paru nécessaire pour assurer la qualité de l'annotation. De plus, puisque toutes les données ont été annotées par les mêmes annotateurs, cela garantit la cohérence dans les annotations.

Le corpus a été annoté en comportement émotionnel (polarité et *arousal*) et social (accord, dominance, engagement, performance et rapport). De plus, au moment de la création de RECOLA, aucun corpus de données spontanées audiovisuelles et physiologiques n'a été disponible gratuitement dans le domaine de la recherche.

Ainsi, le corpus RECOLA est basé sur les interactions spontanées recueillies à partir de la tâche collaborative réalisée par les groupes de participants de deux personnes. Le corpus est disponible pour la recherche personnelle, à la demande sur le site (référence en bas de page) d'un chercheur universitaire.

3.2. Contexte des recherches effectuées pour le corpus RECOLA

Neumann et Thang Vu (2018) ont utilisé le corpus RECOLA pour la reconnaissance cross-lingue et multilingue des émotions en anglais et en français. Les chercheurs ont appliqué l'ACNN (*Attentive convolutional neural network*) pour réaliser quatre expériences

sur la classification binaire de la polarité et de l'arousal : 1) monolingue - la classification des données d'une langue ; 2) multilingue - la classification des données de deux langues fusionnées ; 3) cross-lingue - l'entraînement est réalisé sur les données d'une langue, mais le test est effectué sur les données d'une autre ; 4) cross-lingue + *fine-tuning* : les modèles entraînés lors de l'expérience 3) sont pris comme input et sont ensuite affinés par 100 échantillons choisis au hasard de la langue cible pour chaque étape de la validation croisée. Les résultats de l'expérience ont été mesurés en moyenne du rappel pour chaque classe. Le meilleur score de polarité (52,3) a été obtenu pour la classification monolingue, et de l'arousal (63,07) pour la classification cross-lingue + *fine tuning*.

Dans l'article de Rejaibi et al. (2020) les classifications de la polarité et de l'arousal ont été réalisées avec le CNN. Les résultats ont montré que le CNN développé sur les spectrogrammes a obtenu l'accuracy de 73,81% pour la polarité et 75,65% pour l'arousal, entraîné sur les MFCC - 71,12% pour la polarité et 70,23% pour l'arousal. La combinaison des spectrogrammes et des MFCC a donné des meilleurs résultats d'accuracy - 91,44% pour la polarité et 89,3% pour l'arousal.

Le corpus RECOLA a été également utilisé pour tester la performance des nouvelles méthodes de la classification. Ainsi, Trigeorgis et al. (2016) ont proposé la méthode d'entraînement des modèles à partir du signal acoustique sans extraction des descripteurs pour l'apprentissage. Cette méthode a représenté la combinaison des algorithmes CNN et LSTM (*Long short-term memory*) et a pris comme input les séquences audio de 6 secondes. Le modèle a été testé sur la classification de la polarité et de l'arousal du corpus RECOLA et a été comparé avec les algorithmes SVR (*Support vector regression*) et BLSTM (*Bidirectional long short-term memory*) qui ont été entraînés sur l'ensemble des descripteurs décrit dans le Chapitre 2.4 eGeMAPS (Eyben at al., 2016) et les descripteurs utilisés pour les défis computationnels paralinguistiques ComParE (Schuller et al., 2013). L'approche proposé a dépassé les performances de ces deux algorithmes pour la polarité et l'arousal.

Tzirakis et al. (2018) ont proposé un outil de modélisation *end-to-end*, c'est-à-dire le modèle peut prendre comme input les données audios, vidéos ou physiologiques. Le corpus RECOLA a été également utilisé pour tester la performance de cet outil. Les expériences ont indiqué que l'outil proposait les résultats comparables aux algorithmes les plus performants sans extraction des descripteurs.

Yang et Hirschberg (2018) ont découvert que l'utilisation des audios et des spectrogrammes comme input à l'architecture hybride CNN (Convolutional neural network) + BLTSM (*Bidirectional long short-term memory*) améliore la performance en termes de CCC (*Concordance correlation coefficient*) du système par rapport aux résultats de la même architecture entraînée uniquement soit sur des audios soit sur des spectrogrammes. Le modèle a été testé sur la classification de la polarité et de l'arousal.

3.3. Pré-traitement

Pour ce travail, nous utilisons uniquement l'annotation en polarité. L'annotation du corpus RECOLA est réalisée en continu avec des intervalles de 0.04 secondes. La durée totale du corpus est de 115 minutes. Le pré-traitement des données pour l'entraînement des modèles consiste en deux étapes : la segmentation des données audio, l'extraction des MFCC et la création des images à partir des segments de l'étape précédente.

3.3.1. Segmentation des audios

La partie du dossier du corpus RECOLA disponible pour la recherche contient des fichiers audios et des fichiers tabulaires correspondants aux fichiers audios. L'annotation en polarité présentée dans les fichiers tabulaires est notée entre -1 (négatif) et 1 (positif) avec 0 comme neutre d'un pas de 0,01. Les valeurs entre -0,09 et 0,09 paraissent très faibles pour représenter les polarités négative et positive. Ainsi, on considère que les valeurs de l'annotation entre -1 et -0,1 correspondent aux segments avec la polarité négative, les valeurs entre -0,09 et 0,09 correspondent aux segments avec la polarité neutre, et enfin, les valeurs entre les segments 0,1 et 1 correspondent aux segments avec la polarité positive. Les pas de 0,04 secondes ont été regroupés manuellement dans les segments. Les audios ont été extraits grâce aux valeurs temporelles de ces segments.

L'extraction des fichiers audios est réalisée avec le logiciel Praat (Boersma & Weenink, 2020). Les fichiers audios ont été tout d'abord segmentés en fonction de l'annotation de la polarité décrite précédemment. Si le segment avait une durée inférieure à 2 secondes, ce segment n'était pas gardé. Les extraits sont sauvegardés dans les dossiers correspondants à leur étiquette : neutre, positif et négatif.

Ces extraits sont de durée différente, il a été donc nécessaire de les segmenter en plus petits échantillons. Ces échantillons doivent répondre à deux critères. Premièrement, ils doivent présenter la voix de la personne proche du microphone (participant de l'expérience). Deuxièmement, la durée d'échantillon ne peut pas être inférieure à 2 secondes et ne peut pas dépasser 2,5 secondes.

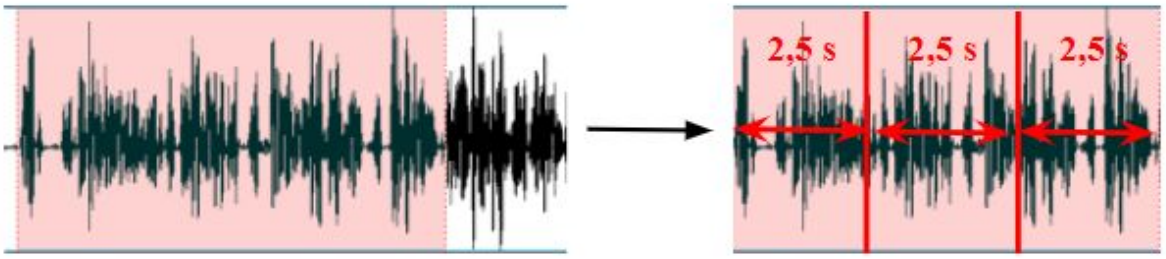

Image 7. Illustration du découpage en extraits courts

L'enregistrement du corpus a été effectué en binômes, mais un fichier audio correspond à la voix d'un seul participant du groupe. Ainsi, il paraît que l'annotation neutre correspond dans la plupart des cas à la voix de l'interlocuteur de la personne enregistrée ou au bruit.

La qualité de l'enregistrement de la voix de la personne au fond n'est pas suffisante pour l'entraînement des modèles d'apprentissage. Sur l'image présentée ci-dessous, la partie du signal acoustique et du spectrogramme en rose correspond à la voix du deuxième participant de l'enregistrement, la ligne bleue à la fréquence fondamentale. On peut constater que la personne est placée très loin du microphone, puisqu'il n'y a pas de ligne de fréquence fondamentale sur le spectrogramme. Les MFCC extraits à partir de ce type du signal ne sont pas représentatifs.

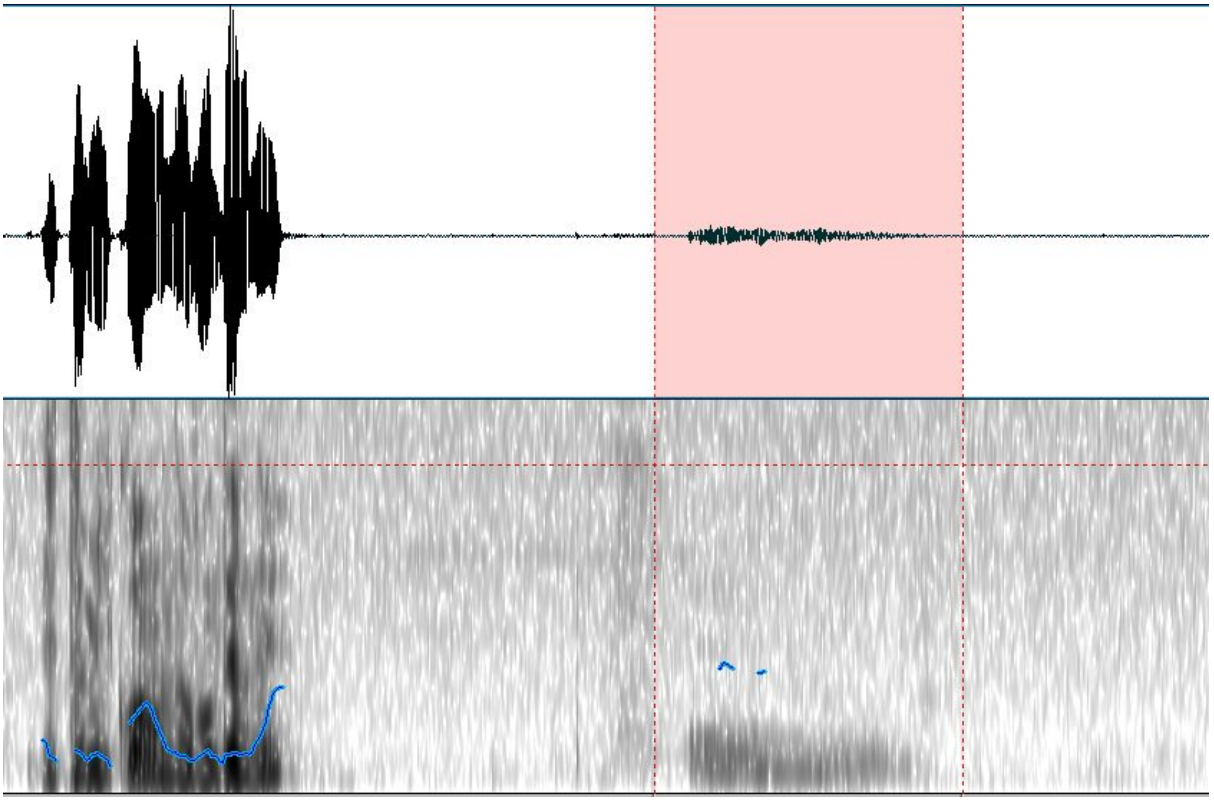

Image 8. Exemple du bruit et de la voix de l'interlocuteur sur le spectrogramme

De plus, lors de l'annotation du corpus, le comportement émotionnel a été annoté uniquement pour le participant, sans tenir compte de la deuxième personne. Ainsi, nous avons décidé de garder uniquement les parties de l'enregistrement avec la voix de la personne principale de l'enregistrement.

Le choix de la durée d'échantillonnage est arbitraire. Après plusieurs essais de paramétrage différent de la durée sur une petite quantité des données, les modèles entraînés sur les extraits de 2 à 2,5 secondes ont montré le meilleur résultat.

La plus grande partie des audios est annotée en neutre, mais comme expliqué précédemment, le neutre correspond au bruit ou à la voix de la personne loin du microphone.

Constitution du corpus pour l'apprentissage et le test des modèles		
classe négative	classe neutre	classe positive
107 extraits	800 extraits	826 extraits

Tableau 1. Constitution du corpus pour l'apprentissage et le test des modèles

3.3.2. Extraction des MFCC

L'extraction des MFCC est réalisée avec la librairie *librosa* (McFee et al., 2015) pour Python. Les premiers 12-20 coefficients cepstraux représentent la voix humaine et sont suffisants pour l'analyse du signal. Les coefficients de plus haut niveau représentent l'information spectrale plus détaillée. L'extraction d'un grand nombre de coefficients cepstraux rend les modèles plus complexes, ce qui nécessite l'utilisation de plus grandes quantités de données. Par convention, nous avons donc extrait les 13 premiers coefficients cepstraux.

L'extraction des MFCC doit être effectuée à partir des fenêtres glissantes. De cette façon, les 13 coefficients cepstraux sont calculés dans les fenêtres de 25 ms avec un chevauchement des fenêtres de 10 ms. Le taux d'échantillonnage des audios n'a pas été changé, il est donc à 44100 kHz. Les extraits de durée de moins 2,5 secondes ont été complétés par la technique de zero-padding. Dans le cas des MFCC, les valeurs manquantes des extraits plus courts peuvent être remplacées par la valeur maximale ou la valeur minimale de MFCC de tout le corpus. De cette façon, tous les extraits analysés sont de même taille, mais la résolution n'est pas changée. Nous avons utilisé la valeur maximale des MFCC, qui correspond aux pixels blancs sur les images.

On a également appliqué la procédure de normalisation par la standardisation (Z-score normalisation). Cette technique permet de représenter l'éloignement d'une valeur par rapport à la moyenne de l'ensemble des données. Pour son calcul, nous avons obtenu la moyenne et l'écart-type pour chacun des 13 coefficients MFCC. Ensuite, on a soustrait de chaque donnée la moyenne et le résultat a été divisé par l'écart-type. La normalisation des données permet de tenir compte de tous les coefficients MFCC indépendamment de leur plage de variation.

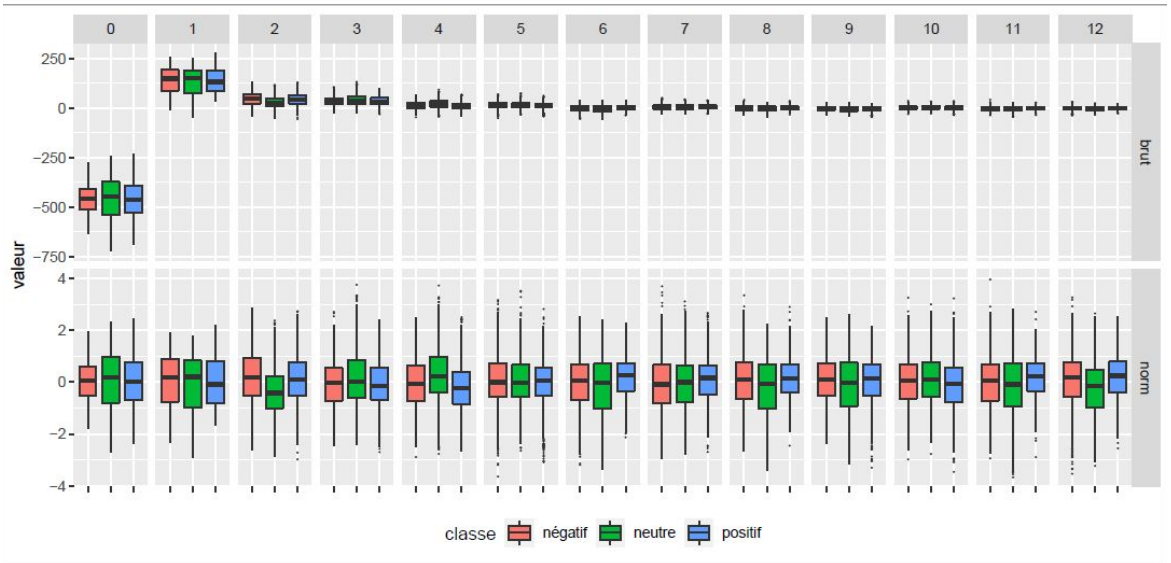

Image 9. Distribution des valeurs de MFCC brutes vs. normalisées

Sur l'image ci-dessus les valeurs brutes des coefficients MFCC pour les trois classes sont représentées sur la première ligne et les valeurs normalisées - sur la deuxième ligne. On observe que la différence entre les valeurs brutes des deux premiers coefficients est très importante, alors que la plage de variation des 11 suivants est beaucoup plus faible. La plus grande partie de l'information présentée par les coefficients suivants est donc perdue. L'image des valeurs de MFCC pour un extrait présentée ci-dessous montre que l'information spectrale se résume principalement en deux lignes horizontales, qui correspondent aux deux premiers coefficients MFCC. En comparaison, les images générées à partir des valeurs normalisées permettent mieux de représenter la variation des 13 coefficients.

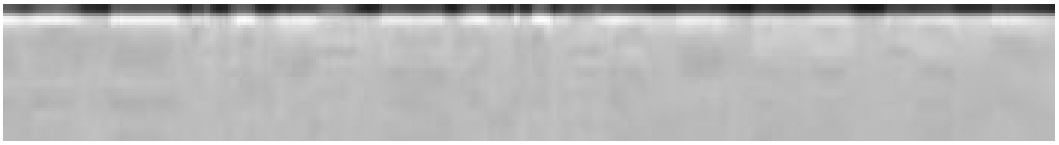

Image 10. Image générée à partir des valeurs brutes des MFCC pour un extrait audio. L'axe horizontal correspond au temps (ici 2,5 s) et l'axe vertical aux 13 coefficients MFCC, du haut vers le bas

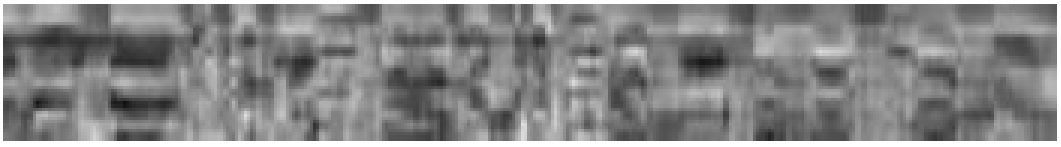

Image 11. Image générée à partir des valeurs normalisées des MFCC pour le même extrait audio que celui représenté par l'image 10

Pour le CNN à deux dimensions les images ont été construites à partir des valeurs normalisées, où les valeurs manquantes des extraits plus courts ont été complétées par les pixels blancs. Les nuances de gris indiquent l'intensité des valeurs.

Image 12. MFCC d'un extrait de moins 2,5 s (technique de zero-padding)

Pour le CNN à une dimension et le SVM les valeurs normalisées des coefficients MFCC de chaque extrait ont été concaténées sur une ligne. De cette façon, nous avons obtenu le tableau de 1733 lignes (nombre d'extraits total) et de 2169 colonnes (13 coefficients * 167 pas de 25 ms). Pour que tous les extraits aient le même format, les extraits courts ont été complétées par la valeur maximale de MFCC de tout le corpus.

Les statistiques des 13 coefficients MFCC ont été extraites pour les fenêtres de 25 ms. La moyenne, l'écart-type, la moyenne de la dérivée première (delta) et la moyenne de la dérivée seconde (delta-delta) ont été calculés pour chacun des 13 coefficients sur l'ensemble de l'extrait audio. La dérivée première indique la variation entre les fenêtres consécutives. La dérivée seconde (différences entre différences) est supposée donner une idée de la variation à plus longue terme. Ces valeurs ont été concaténées dans un tableau, où chaque extrait est décrit par 52 vecteurs (13 coefficients * 4 mesures).

3.4. SVM

Les modèles SVM ont été développés avec la bibliothèque *sklearn* de python. Pour la construction de l'algorithme nous avons procédé par les étapes suivantes : les valeurs de MFCC et les étiquettes des extraits sont sauvegardées dans les variables et séparées au hasard en ensembles d'apprentissage (75%) et de test (25%). Ensuite l'algorithme SVC (*Support Vector Classification*) est appliqué et le modèle est testé sur l'ensemble de test.

Cet algorithme réalise la classification pour plusieurs classes avec l'approche *one-vs-one*. Cette approche réalise la classification binaire de chaque classe avec une autre classe. Dans notre cas, les classifications binaires sont les suivantes : négatif - neutre, négatif-positif, neutre-positif. Ce modèle est entraîné avec le kernel poly de degré 3, qui est la fonction de noyau polynomiale qui permet de modéliser les données non-linéaires.

3.4.1. SVM entraîné sur les statistiques de MFCC

Ce modèle a été entraîné sur la moyenne, l'écart-type, la dérivée première moyenne et la dérivée seconde moyenne des coefficients MFCC sur toute la durée de l'extrait.

	précision	rappel	f-mesure	f-mesure globale	accuracy
négatif	0,0	0,0	0,0	0,43	0,63
neutre	0,62	0,67	0,65		
positif	0,65	0,66	0,65		

Tableau 2. Résultats du SVM entraîné sur les statistiques

Les résultats de ce modèle ont montré que la classification est effectuée correctement dans 63% des cas. La précision de la classe neutre est de 0,62, le rappel est de 0,67 et la F-mesure est de 0,65. La classe positive a obtenu la précision de 0,65, le rappel de 0,66 et la F-mesure de 0,65. Néanmoins, aucun échantillon de l'apprentissage n'a été étiqueté comme négatif, ce qui baisse le score global de la F-mesure.

3.4.2. SVM entraîné sur les valeurs de MFCC

L'autre modèle SVM a été entraîné sur les valeurs de MFCC extraits des fenêtres glissantes de 25 ms avec 10 ms de chevauchement.

	précision	rappel	f-mesure	f-mesure globale	accuracy
négatif	0,0	0,0	0,0	0,33	0,53
neutre	0,69	0,24	0,35		
positif	0,5	0,9	0,64		

Tableau 3. Résultats du SVM entraîné sur les valeurs de MFCC

Le classifieur entraîné sur les valeurs de MFCC a eu un score global de 53% d'extraits classés correctement. Pour la classe neutre, la précision est de 0,69, le rappel est de 0,24 et la F-mesure est de 0,35. La classe positive a obtenu la précision de 0,5, le rappel de 0,9 et la

F-mesure de 0,64. De même que le modèle entraîné sur les statistiques, ce modèle n'a classé aucun extrait comme négatif.

3.5. CNN

Les modèles CNN à une dimension ont été également développés sur les statistiques de MFCC calculés pour toute la durée de chaque extrait et sur les valeurs de MFCC extraites à partir des fenêtres glissantes de 25 ms. Puisqu'on a utilisé le même format des données que celui pour les modèles SVM, la comparaison des performances des algorithmes est plus directe. De plus, le CNN à une dimension permet de tester l'approche de Li et al. (2019). Les résultats de cette approche montrent que les CNN à une dimension donnent des résultats comparables à ceux des CNN à deux dimensions pour le traitement de la parole.

Nous avons également appliqué l'approche plus traditionnelle qui est le développement du CNN à deux dimensions sur les images. Les images ont été générées sur les valeurs de MFCC des fenêtres glissantes de 25 ms avec un chevauchement de 10 ms.

Pour les deux types de modèles, les CNN sur Python ont été réalisés avec la bibliothèque *Keras* (Chollet, 2015).

3.5.1 CNN à une dimension

Les modèles CNN entraînés sur les valeurs de MFCC et sur les statistiques de MFCC ont la même structure. Après le pré-traitement, les données ont été séparées au hasard en ensemble d'entraînement et en ensemble de test, en gardant les mêmes proportions des ensembles que pour l'apprentissage du SVM. Ainsi, l'ensemble d'apprentissage contient 75% des données et l'ensemble de test - 25%. Ensuite, les valeurs de MFCC et leurs étiquettes sont converties au format array de la bibliothèque *numpy* (Travis, 2006). Les valeurs ont été transformées en matrice à une dimension. Dans l'exemple présenté ci-dessous, les valeurs entre deux crochets correspondent aux valeurs de MFCC d'un extrait et toutes les données de l'ensemble de test (et d'entraînement) sont mises entre les crochets en gras :

```
[[[ 1.4526689 ]
  [ 1.4212904 ]
  [-0.84215796]
      ...
```

```
 [-0.02284028]
 [-0.8502754 ]
[-0.7503511 ]]
      ...
[[-1.8981771 ]
 [-1.4891632 ]
 [-0.19543904]
      ...
 [ 5.81137129]
 [ 5.81137129]
[ 5.81137129]]]
```

Les étiquettes ont été d'abord mis sous la forme numérique et ensuite ont été représentées avec l'encodage one-hot. Par exemple, l'étiquette "neutre" est devenue "0" et ensuite [1. 0. 0.]. L'encodage one-hot prend la forme du nombre de classes dans le corpus et donne à chaque classe un indice qui est représenté par le chiffre 1.

Le CNN est entraîné sur le modèle *Sequential*. Les couches de convolution ont été construites avec la fonction d'activation *relu* et les couches de pooling avec la fonction max-pooling. Nous avons également ajouté des couches de normalisation et des couches *dropout*. Les couches de normalisation servent à normaliser les activations de la couche précédente afin d'éviter la variation de la distribution des descripteurs à travers des données de l'entraînement et du test. Ces couches appliquent la transformation qui permet de garder la moyenne de l'activation à 0 et mesurer l'activation en écart-type. Les couches *dropout* ont pour but de rendre le modèle moins complexe afin de prévenir l'*over-fitting* du système par la désactivation de quelques neurones au hasard lors de l'apprentissage. Une autre couche utilisée est *Flatten* qui transforme la matrice en un seul rang (*array*). Pour compiler le modèle nous avons utilisé l'optimiseur RMSprop. De cette façon, l'architecture du réseau de neurones convolutif est construit de 6 couches convolutives, 1 couche *dropout*, 2 couches de normalisation, 2 couches de *pooling* et 1 couche *Flatten*. Les modèles ont été testés sur 15 époques avec le batch-size 32.

3.5.1.1. CNN à une dimension entraîné sur les statistiques de MFCC

De même que le modèle SVM, le CNN à une dimension a été entraîné sur la moyenne, l'écart type, la dérivée première moyenne et la dérivée seconde moyenne des 13 coefficients MFCC sur la durée de chaque extrait.

	précision	rappel	f-mesure	f-mesure globale	accuracy
négatif	0,6	0,27	0,37	0,53	0,6
neutre	0,62	0,55	0,58		
positif	0,58	0,68	0,63		

Tableau 4. Résultats du CNN 1d entraîné sur les statistiques de MFCC

Les échantillons ont été étiquetés correctement dans 60% par ce modèle avec la f-mesure globale de 0,53. Le rappel de la classe neutre est de 0,55, de la classe positive est de 0,68 et de la classe négative est de 0,27. La classe négative a obtenu le score de la précision de 0,6, la classe neutre - 0,62 et la classe positive - 0,58. Il est remarquable qu'aucun extrait annoté en positif par les annotateurs humains n'a pas été identifié comme négatif par le modèle.

3.5.1.2. CNN à une dimension entraîné sur les valeurs de MFCC

Les valeurs de MFCC extraites des fenêtres glissantes ont été concaténées sur une ligne pour chaque extrait. Les valeurs manquantes des extraits plus courts ont été remplacés par la valeur maximale de tout le corpus.

	précision	rappel	f-mesure	f-mesure globale	accuracy
négatif	0,5	0,14	0,21	0,47	0,59
neutre	0,58	0,66	0,62		
positif	0,61	0,57	0,58		

Tableau 5. Résultats du CNN 1d entraîné sur les valeurs de MFCC

Les extraits ont été étiquetés correctement dans 59% des cas avec la F-mesure globale de 0,47. Ainsi, la classe négative a obtenu la précision de 0,5, le rappel de 0,14 et la f-mesure de 0,21. La précision de la classe neutre est de 0,58, le rappel est de 0,66 et la F-mesure de 0,62. La classe positive a eu le score 0,61 de précision, 0,57 de rappel et 0,58 de F-mesure.

3.5.2. CNN à deux dimensions

L'approche utilisée pour les autres modèles pour la séparation du corpus en ensemble d'apprentissage et de test peut être appliquée sur les données au format tabulaire. Pour réaliser la séparation des images en ensembles d'entraînement et de test, nous avons récupéré les noms des extraits de chaque classe, avons séparé au hasard cette liste en proportions de 75% pour l'ensemble d'apprentissage et 25% pour l'ensemble de test et avons copié les images dans les dossiers correspondant aux ensembles (voir Annexe 5).

Pour l'apprentissage du modèle CNN avec les images nous avons utilisé l'application Google Colab qui permet d'appliquer l'accélérateur GPU. Le GPU optimise le travail avec les données graphiques.

Le dossier des ensembles d'apprentissage et de test a été zippé et a été mis sur Google Drive. Le dossier a été ensuite dézippé dans un dossier de Google Drive.

Les images ont été chargées et ont été redimensionnées à la même taille de 28*28. Nous avons testé d'autres configurations comme l'application d'une taille plus grande ou l'utilisation des tailles réelles des images. Ces modèles ont été moins performants et le temps de traitement a été augmenté.

Les données des images sont ensuite transformées au format adapté de numpy, redimensionnées aux deux dimensions et sont normalisées pour garder la même intensité de couleur. Les étiquettes sont transformées avec l'encodage one-hot.

Le réseau de neurones est construit en deux couches de convolution avec le filtre de taille 3x3, une couche de *pooling*, deux couches de *dropout*, deux couches *dense* et une couche *flatten*.

Les résultats du test du modèle sont sauvegardés et ils sont transformés à l'encodage one-hot pour pouvoir les comparer avec l'ensemble de test.

	précision	rappel	f-mesure	f-mesure globale	accuracy
négatif	0,33	0,11	0,17	0,44	0,56
neutre	0,55	0,56	0,56		
positif	0,58	0,62	0,6		

Tableau 6. Résultats du CNN 2d entraîné sur les images des valeurs de MFCC

Ainsi, le modèle à deux dimensions entraîné sur les images a eu la F-mesure globale de 0,44 et l'accuracy de 0,56. La précision de la classe négative est de 0,33, le rappel est de 0,11 et la F-mesure est de 0,17. La classe neutre a obtenu la précision de 0,55, le rappel de 0,56 et la F-mesure de 0,56. Pour la classe positive la précision est de 0,58, le rappel est de 0,62 et la F-mesure est de 0,6.

3.6. Discussion

Dans ce travail, nous avons développé des modèles SVM et CNN pour la classification automatique de la parole en polarité positive, neutre et négative. Le SVM et le CNN à une dimension ont été entraînés sur les statistiques de MFCC de tout le signal et sur les valeurs de MFCC extraites des fenêtres glissantes de 25 ms. Le CNN à deux dimensions a été créé sur les images construites à la base des valeurs de MFCC des fenêtres glissantes. Le tableau récapitulatif des performances des 5 modèles est présenté ci-dessous.

		précision	rappel	f-mesure	f-mesure globale	accuracy
SVM entraîné sur les statistiques de MFCC	négatif	0,0	0,0	0,0	0,43	**0,63**
	neutre	0,62	0,67	**0,65**		
	positif	0,65	0,66	**0,65**		
SVM entraîné sur les valeurs de MFCC	négatif	0,0	0,0	0,0	0,33	0,53
	neutre	0,69	0,24	0,35		
	positif	0,5	0,9	0,64		
CNN 1d entraîné sur les statistiques de MFCC	négatif	0,6	0,27	**0,37**	**0,53**	0,6
	neutre	0,62	0,55	0,58		
	positif	0,58	0,68	0,63		
CNN 1d entraîné sur les valeurs de MFCC	négatif	0,5	0,14	0,21	0,47	0,59
	neutre	0,58	0,66	0,62		
	positif	0,61	0,57	0,58		
CNN 2d entraîné sur les images MFCC	négatif	0,33	0,11	0,17	0,44	0,56
	neutre	0,55	0,56	0,56		
	positif	0,58	0,62	0,6		

Tableau 7. Récapitulatif des résultats des modèles

Le SVM entraîné sur les statistiques prédit le mieux les classes positive et neutre. Le SVM développé sur les valeurs de MFCC retourne 358 sur 434 échantillons étiquetés comme positif sur tout l'ensemble de test. Les deux modèles n'ont identifié aucun extrait négatif.

Le CNN à une dimension entraîné sur les statistiques de MFCC reconnaît le mieux la polarité négative parmi tous les modèles. Ce modèle est également performant sur la détection de la polarité positive. Le CNN à une dimension entraîné sur les valeurs de MFCC, au contraire, détecte mieux la polarité neutre que la polarité positive.

Le CNN est traditionnellement appliqué sur les données graphiques. Dans ce mémoire le modèle à deux dimensions entraîné sur les images est moins performant pour la reconnaissance des polarités négative et neutre que le modèle à une dimension entraîné sur les mêmes valeurs.

Contrairement aux SVM, tous les modèles CNN sont capables de distinguer la polarité négative, même si le résultat reste très bas. La plupart des extraits de la polarité négative (12 sur 22) ont été identifiés comme neutre par le modèle à une dimension entraîné sur les statistiques de MFCC. De plus, ces trois modèles ont les scores de rappel, de précision et de F-mesure plus équilibrés.

On peut conclure que le SVM est peu adapté à la reconnaissance de la polarité dans le cas quand le corpus est déséquilibré. Le CNN est plus performant dans ce cas, malgré des résultats qui restent bas. Parmi tous les CNN, le modèle à une dimension entraîné sur les statistiques de MFCC a montré le meilleur score pour la reconnaissance de la polarité négative et de la polarité positive. Ce modèle a également le meilleur score de F-mesure de tous les modèles. Ces résultats montrent que dans le cas quand le corpus est déséquilibré et les extraits n'ont pas la même durée le modèle le plus adapté est le CNN à une dimension entraîné sur les statistiques de MFCC. Malgré le fait que le volume des corpus utilisés dans le travail de Li et al. (2019) est beaucoup plus important que la taille du corpus utilisé dans ce mémoire, les résultats du CNN à une dimension entraîné sur les statistiques sont comparables avec ceux de Li et al. (2019) pour le corpus IEMOCAP.

L'importance des statistiques de MFCC comme des descripteurs pour le développement des modèles pour la reconnaissance des émotions peut être étudiée par la comparaison des classifieurs entraînés sur des vecteurs de valeurs. Les combinaisons différentes des valeurs utilisées dans ce mémoire (la moyenne, l'écart-type, la dérivée première et la dérivée seconde) permettent de savoir quelle valeur apporte le plus d'information sur le contenu émotionnel.

Plusieurs explications peuvent être données pour ces résultats. Premièrement, cela peut être dû au choix de l'interprétation de l'annotation qui a été faite lors de la segmentation. Ainsi, il est possible que le seuil choisi pour le neutre (de -0,09 à 0,09) soit très large et ne représente donc pas la polarité neutre. De cette façon, les extraits annotés proche de 0,1 pourraient être identifiés comme positifs. Il n'est pas possible de vérifier cette hypothèse, puisque le format des données passées à l'input du CNN et du SVM contenait uniquement les valeurs des descripteurs d'apprentissage et les labels. La valeur numérique de l'annotation de chaque extrait n'a pas été sauvegardé lors du pré-traitement.

Les valeurs limites de la séparation des valeurs d'annotation auraient pu être défini en fonction de la distribution des valeurs dans le corpus. Dans ce cas, le corpus aurait pu être plus équilibré.

Deuxièmement, comme déjà mentionné dans la partie théorique de ce mémoire, dans le domaine de l'analyse des sentiments, il existe deux façons de développer des modèles : garder les proportions de la présentation de la polarité dans la parole ou équilibrer les données dans le corpus d'apprentissage. Nous avons préféré garder les proportions de la polarité présentée dans le corpus RECOLA. La particularité du corpus d'apprentissage a été la dominance de la polarité positive sur la polarité neutre et la petite quantité des données annotées en polarité négative. Une des façons d'équilibrer les proportions des données dans un corpus d'apprentissage, c'est l'utilisation de plusieurs corpus dont les guides d'annotation et/ou d'enregistrement suivent le même schéma.

Troisièmement, la partie du corpus RECOLA disponible pour la recherche ne contient que 115 minutes de données. Lors de la segmentation, la taille du corpus a diminué : les parties contenant uniquement le bruit de l'environnement sans la voix du participant ont été éliminés. De plus, le corpus a été découpé en segments de minimum 2 secondes, si la suite de l'annotation d'une polarité était de moins de 2 secondes, cette partie de l'enregistrement a été également enlevée. Ainsi, le corpus final d'apprentissage n'est pas suffisant pour l'apprentissage automatique.

Conclusion

Le but de ce mémoire était d'établir quel classifieur et quels descripteurs sont les plus adapté pour réaliser la classification automatique de la polarité à l'oral. Dans cet objectif nous avons comparé les performances de deux algorithmes d'apprentissage, le SVM et le CNN, entraîné sur des statistiques et des valeurs de MFCC.

Dans un premier temps, nous avons étudié les théories des émotions, afin d'en choisir une pour la modélisation des classifieurs. Parmi toutes les approches étudiées, la perspective unidimensionnelle a été sélectionnée pour sa performance dans le domaine de l'analyse des sentiments.

Dans un second temps, nous avons détaillé le domaine de l'analyse des sentiments, et plus précisément les cas d'usages, les problématiques et les descripteurs d'apprentissage. Notre corpus d'apprentissage ne contient pas de transcription, nous nous sommes donc concentrés sur les descripteurs non-linguistiques. Parmi ces descripteurs, un des plus utilisés est le MFCC grâce aux critères statistiques et leurs bonnes performances dans les tâches de classification.

Nous avons également expliqué le fonctionnement du SVM et du CNN. Le SVM est largement utilisé pour tous types de données, notamment les données audio. Le CNN est connu pour sa très haute performance au niveau de la reconnaissance des images. La performance des CNN à une dimension est peu étudiée pour la classification des émotions à l'oral. Ainsi, nous avons repris l'idée de l'architecture du CNN à une dimension de Li et al. (2019) pour l'appliquer à la classification de la polarité.

Le corpus RECOLA a été segmenté en classes positif, neutre et négatif par les extraits de 2 à 2,5 secondes. Les valeurs de MFCC ont été calculés à partir des fenêtres glissantes de 25 ms avec la durée de chevauchement de 10 ms et elles ont été normalisées. Les valeurs manquantes pour les extraits plus courts ont été remplacées par la valeur maximale de MFCC de tout le corpus. Ces valeurs ont été utilisées comme descripteurs pour l'entraînement du SVM et du CNN à une dimension. De plus, les images ont été générées à la base de ces valeurs pour l'entraînement du CNN à deux dimensions. La moyenne, l'écart-type, la dérivée première moyenne et la dérivée seconde moyenne ont été calculés pour chaque extrait. Ces

valeurs ont été utilisées comme des descripteurs pour l'entraînement du SVM et du CNN à une dimension.

Les statistiques de MFCC illustrent les valeurs globales des extraits, les valeurs de MFCC permettent d'analyser l'évolution des coefficients MFCC au cours du temps. L'utilisation traditionnelle du CNN à deux dimensions rend possible l'évaluation du CNN à une dimension appliquée au traitement de l'oral.

Nous avons découvert que les modèles CNN sont mieux adaptés pour la reconnaissance de la polarité dans le cas du corpus déséquilibré et des extraits à durée différente que le SVM. Le CNN à une dimension entraîné sur les valeurs de MFCC a montré de meilleurs résultats que le CNN à deux dimensions développé sur les images. L'approche de Li et al. (2019) a donc prouvé son efficacité. En tenant compte des scores de rappel, de précision, de F-mesure de chaque classe, ainsi que de F-mesure globale et d'accuracy, le modèle le plus performant pour la détection de la polarité à l'oral est le CNN à une dimension entraîné sur la moyenne, l'écart-type, la dérivée première moyenne et la dérivée seconde moyenne des MFCC.

Bibliographie

Allport, G.W. (1935). Attitude, extrait d'un article In *Handbook of Social Psychology*, ed. Murchison, Clark University Press, Worcester.

Anagnostopoulos, C. -N., Iliou, T., & Giannoukos, I. (2015). Features and classifiers for emotion recognition from speech: A survey from 2000 to 2011. *Artificial Intelligence Review*, 43(2), 155–177.

Averill, J.R. (1980). A constructivist view of emotions. Dans R. Plutchik et H. Kellerman (dir.), *Emotion : Theory, research and experience*, vol.1 (p. 305-339). New York : Academic Press.

Arnold, M. B. (1960). *Emotion and Personality*. New York:Columbia University Press.

Audiffren, J. (2017). https://dataanalyticspost.com/Lexique/svm/

Banse, R., Scherer, K. (1996). Acoustic profiles in vocal emotion expression. *Journal of Personal and Social Psychology,* 70(3), 614–636.

Bard, P. (1928). A diencephalic mechanism for the expression of rage with special reference to the central nervous system. *American Journal of Physiology*, 84, 490-513.

Batliner, A., Fischer, K., Huber, R., Spilker, J., Noth, E. (2003). How to find trouble in communication. *Speech Communication,* 40, 117–143.

Batliner, A., Fischer, K., Huber, R., Spilker, J., Noth, E. (2000). Desperately Seeking emotions : actors, wizards and human beings. *Proceedings the ISCA Workshop Speech Emotion*, 195–200.

Boersma, P., Weenink, D. (2020). Praat: doing phonetics by computer [Computer program]. Version 6.1.16, retrieved 6 June 2020 from http://www.praat.org/

Borchert, M., Dusterhoft, A. (2005). Emotions In Speech - experiments with prosody and quality features in speech for use in categorical and dimensional emotion recognition environments. *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, 147–151.

Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., Weiss, B. (2005). A Database Of German emotional speech. *Proceedings of the Interspeech 2005*, 1517–1520.

Busso, C., Bulut, M., Chun Lee, C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., Narayanan, S. S. (2008). Iemocap: interactive emotional dyadic motion capture database, language resources and evaluation.

Cahn, J. (1990). The generation of affect in synthesized speech. *Journal of the American Voice Input Output Society*, 8, 1–19.

Cannon, W. B. (1927). The James-Lange theory of emotions: A critical examination and an alternative theory. *American Journal of Psychology*, 39, 106-124.

Chollet, F. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Cohen, R. (1984). A computational theory of the function of clue words in argument understanding. *ACL-22:Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting on Association for Computational Linguistics,* 251–258.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Kollias, S., Fellenz, W., Taylor, J. (2001). Emotion recognition in human–computer interaction. *IEEE Signal Processing*, 18, 32–80.

Darwin, C. (1872). *The expression of the emotions in man and animals*. London: John Murray.

Dumas, G. (1948). *La Vie affective, Physiologie - Psychologie - Socialisation.* Paris: P.U.F.

Ekman, P. (1972). Universals and cultural differences in facial expression of emotion. Lincoln: University of Nebraska Press.

El Ayadi, M., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3), 572–587.

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., et al. (2016). The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE Transactions on Affective Computing*, 7, 190–202.

Eyben, F., Weninger, F., Schuller, B. (2013). Affect recognition in real-life acoustic conditions. A new perspective on feature selection. *Proceedings of the Interspeech 2013, 14th Annual Conference of the International Speech Communication Association*.

Ferragne, E., Gendrot, C., Pellegrini, T. (2019). Towards phonetic interpretability in deep learning applied to voice comparison. *ICPhS*, Melbourne, Australia. pp.ISBN 978-0-646-80069-1. ffhalshs-02412948f

Fontaine, J. R. J., Scherer, K. R., Roesch, E. B., Ellsworth, P. C. (2007). The world of emotions is not two dimensional. *Psychological Science*, 18, 1050–1057.

Forbes-Riley, K., Litman, D. J. (2004) Predicting emotion in spoken dialogue from multiple knowledge sources. *Proceedings of HLT/NAACL*, 201–208.

France, D. J., Shiavi, R.G., Silverman, S., Silverman, M., Wilkes, M. (2000). Acoustical properties of speech as indicators of depression and suicidal risk. *IEEE Transactions on Biomedical Engineering*, 47(7), 829–837. doi: 10.1109/10.846676

Frank, E., Hall, M. A., Witten, I. H. (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition.

Gergen, K. J. (1985). The social constructionist movement in modern psychology. *American Psychologist,* 40, 266-275.

Gobl, C., Chasaide, A. N. (2003). The role of voice quality in communicating emotion, mood and attitude. *Speech Communication,* 40(1–2), 189–212.

Grosz, B. J., Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3), 175–204.

Gunes, H., Pantic, M. (2010). Automatic, dimensional and continuous emotion recognition. *International Journal of Synthetic Emotions*,1, 68–99.

Gunes, H., Schuller, B., Pantic, M., Cowie, R. (2011). Emotion representation, analysis and synthesis in continuous space: A survey. *Proceedings of the Face and Gesture*, 827–834.

Hansen, J., Cairns, D. (1995). Icarus : source generator based real-time recognition of speech in noisy stressful and Lombard effect environments. *Speech Communication*, 16 (4), 391–422.

Hu, H., Xu, M.-X., Wu, W. (2007). Fusion of global statistical and segmental spectral features for speech emotion recognition. *Interspeech2007*, 2, 1013–1016.

Ijima, Y., Tachibana, M., Nose, T., Kobayashi, T. (2009). Emotional speech recognition based on style estimation and adaptation with multiple-regression HMM. *Proceedings of 2009 IEEE international conference on acoustics, speech and signal processing*, 4157–4160.

Izard, C. E. (1971). *The face of emotion.* Appleton Century Crofts, New York.

James, W. (1890). *The principles of psychology*. Londres: Methuen.

Jeon, J. H., Xia, R., Liu, Y. (2011). Sentence level emotion recognition based on decisions from subsentence segments. *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, 4940–4943.

Johnstone, T., VanReekum, C. M., Hird, K., Kirsner, K., Scherer, K. R. (2005). Affective speech elicited with a computer game, *Emotion*, 5(4), 513–518.

Kanakaraj, M., Guddeti, R. M. R. (2015). NLP based sentiment analysis on Twitter data using ensemble classifiers. *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015, 1-5, doi: 10.1109/ICSCN.2015.7219856.

Ke, X., Zhu, Y., Wen, L., Zhang, W. (2018). Speech emotion recognition based on SVM and ANN. *International Journal of Machine Learning and Computing*, 8(3), 198-202.

Keras Conv1D: Working with 1D Convolutional Neural Networks in Keras. (s.d.) https://missinglink.ai/guides/keras/keras-conv1d-working-1d-convolutional-neural-networks-keras/

Kim, J., Lee, S., Narayanan, S. S. (2010). An exploratory study of manifolds of emotional speech. *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 5142–5145.

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

Lange, C. (1895). *Les émotions*, trad. G. Dumas, Paris, Alcan.

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on International Conference on Machine Learning*, 32, 1188-1196.

Lee, C. M., Narayanan, S. S. (2005). Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13, 293–303.

Lee, C., Yildrim, S., Bulut, M., Kazemzadeh, A., Busso, C., Deng, Z., Lee, S., Narayanan, S. (2004). Emotion Recognition Based On Phoneme Classes. *Proceedings of ICSLP*, 2193–2196.

Leinonen, L., Hiltunen, T. (1997). Expression Of Emotional-motivational connotations with a one-word utterance. *Journal of the Acoustical Society of America*, 102(3), 1853–1863.

Li, Y., Baidoo, C., Cai, T., Kusi, G. A. (2019). Speech Emotion Recognition Using 1D CNN with No Attention. *2019 23rd International Computer Science and Engineering Conference (ICSEC)*. doi:10.1109/icsec47112.2019.8974716

Livingstone, S. R., Russo, F. A. (2018). The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLOS ONE, 13*, 1–35.

Lu, Y. (2015). *Etude contrastive de la prosodie audio-visuelle des affects sociaux en chinois mandarin vs. français : vers une application pour l'apprentissage de la langue étrangère ou seconde*. Linguistique. Université Grenoble Alpes.

Lugger, M., Yang, B. (2007). The Relevance of Voice Quality Features in Speaker Independent Emotion Recognition. *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, IV-17–IV-20.

Ma, J., Jin, H., Yang, L., Tsai, V. (2006). *Ubiquitous Intelligence and Computing : Third International Conference*. Springer, Wuhan, China.

Malrieu, P. (1952). Les émotions et la personnalité de l'enfant. *Études de Psychologie et de Philosophie, XII*. Paris Librairie philosophique.

McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., Nieto, O. (2015). librosa: Audio and music signal analysis in python. *Proceedings of the 14th python in science conference*, pp. 18-25.

Mohammad, S. (2016). Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text. *Emotion Measurement,* 201-237.

Morrison, D., Wang, R., DeSilva, L. (2007). Ensemble Methods For Spoken emotion recognition in call-centres. *Speech Communication*, 49(2), 98–112.

Murray, I., Arnott, J. (1993). Toward a simulation of emotions in Synthetic Speech: A review of the literature on human vocal emotion. *Journal of the Acoustical Society of America*, 93(2), 1097–1108.

Neumann, M., Thang Vu, N. (2018). Cross-lingual and Multilingual Speech Emotion Recognition on English and French. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp.2018.8462162

Noordewier, M. K., Breugelmans, S. M. (2013). On the valence of surprise. *Cognition & Emotion,* 27(7), 1326–1334. doi:10.1080/02699931.2013.777660

Nwe, T., Foo, S., DeSilva,L. (2003). Speech emotion recognition using hidden Markov models. *Speech Communication*, 41, 603–623.

Origlia, A., Galatà, V., Ludusan, B. (2010). Automatic classification of emotions via global and local prosodic features on a multilingual emotional database. *Speech Prosody*, paper 213.

Ortony, A., Clore, G. L., Collins, A. (1988). *The cognitive structure of emotions*. Cambridge University Press.

Osgood, C. E. (1966). Dimensionality of the semantic space for communication via facial expressions. *Scandinavian Journal of Psychology*, 7, 1-30.

Oster, A., Risberg, A. (1986). The Identification Of The mood of a speaker by hearing impaired listeners. *Speech Transmission Laboratory. Quarterly Progress Status Report*, 4, 79–90.

Pan, Y., Shen, P., Shen, L. (2012). Speech emotion recognition using support vector machine. *International Journal of Smart Home*, 6(2), 101-108.

Phand, S. A., Phand, J. A. (2017). Twitter sentiment classification using stanford NLP. *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 1-5, doi: 10.1109/ICISIM.2017.8122138.

Picard, R. W., Vyzas, E., Healey, J. (2001). Toward Machine Emotional Intelligence: analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), 1175–1191.

Plitchuk, R. (1984). Emotions : A general psychoevolutionary theory. *Approaches to Emotion*, ed. Scherer, K.R. & Ekman, P., Hillsdale, NJ : Lawrence Erlbaum Associates.

Rabiner, L., Schafer, R. (1978). Digital Processing of Speech Signals. *Pearson Education*.

Rejaibi, E., Kadoch, D., Bentounes, K., Alfred, R., Daoudi, M., Hadid, A., Othmani, A. (2020). Towards Robust Deep Neural Networks for Affect and Depression Recognition. *arXiv: Human-Computer Interaction*.

Ringeval, F., Sonderegger, A., Sauer, J., Lalanne, D. (2013). Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 1-8, doi: 10.1109/FG.2013.6553805.

Rosenberg, M. J., Hovland, C.I. (1960). Cognitive, affective, and behavioral components of attitudes. *Attitude Organization and Change*, eds. C. I. Hovland & M.J. Rosenberg, New Haven and London, Yale University Press.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology, 39*(6), 1161–1178. https://doi.org/10.1037/h0077714

Russell, J. A., Bachorowski, J., Fernandez-Dols, J. (2003). Facial and vocal expressions of emotion. *Annual Review of Psychology*, 54, 329–349.

Schachter, S., Singer, J.E. (1962). Cognitive, social and physiological determinants of emotional state. *Psychological Review*, 69, 379-399.

Scherer, K. R. (1986). Vocal affect expression : A review and a model for future research. *Psychological Bulletin*, 99(2), 143–165.

Scherer, K. R. (2001). Appraisal considered as a process of multilevel sequential checking. Dans K. R. Scherer, A. Schorr, & T. Johnstone (dir.), *Series in affective science. Appraisal processes in emotion: Theory, methods, research* (p. 92–120). Oxford University Press.

Schuller, B., Rigoll, G., Lang, M. (2004). Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. *Proceedings of the ICASSP 2004*, 1, 577–580.

Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Mortillaro, M., Salamin, H., Polychroniou, A., Valente, F., Kim, S. (2013). The INTERSPEECH 2013 Computational Paralinguistics Challenge: Social signals, conflict, emotion, autism. *Proceedings INTERSPEECH*, 148– 152, Lyon, France.

Seehapoch, T., Wongthanavasu, S. (2013). Speech emotion recognition using Support Vector Machines. *2013 5th International Conference on Knowledge and Smart Technology (KST)*. doi:10.1109/kst.2013.6512793

Shami, M. T., Kamel, M.S. (2005). Segment-based approach the recognition emotions in speech. *IEEE International Conference on Multimedia and Expo*. doi: 10.1109/ICME.2005.1521436.

Shochi, T. (2008). *Prosodie des affects socioculturels en japonais, français et anglais : à la recherche des vrais et faux-amis pour le parcours de l'apprenant*. Université de Stendhal - Grenoble III.

Shrawankar, U., Thakare, V. M. (2013). Techniques for feature extraction in speech recognition system: A comparative study. *International Journal of Computer Applied Engineering and Technology*, 1145, 412–418.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the conference on empirical methods in natural language processing*.

Tao, J., Kang, Y., Li, A. (2006). Prosody conversion from neutral speech to emotional speech. *IEEE Transactions on Audio Speech and Language Processing*, 14(4), 1145–1154.

Thomas, W. I., & Znaniecki, F. (1918). *The Polish Peasant in Europe and America, Monograph of an immigrant group*, Chicago, III, University of Chicago Press.

Travis, E. O. (2006).*A guide to NumPy*. USA: Trelgol Publishing.

Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B. W., Zafeiriou, S. (2016). Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. *Proceedings 41st IEEE international conference on acoustics, speech, and signal processing*, 5200–5204.

Tursunov, A., Kwon, S., Pang, H.-S. (2019). Discriminating Emotions in the Valence Dimension from Speech Using Timbre Features. *Applied Sciences,* 9(12), 2470. doi:10.3390/app9122470

Tzikaris, P., Zafeiriou, S., Schuller, B. (2018). End2You – The Imperial Toolkit for Multimodal Profiling by End-to-End Learning. *arXiv:1802.01115v1*

Vallerand, R. J., Lafrenaye, Y. (2006). Les attitudes. In R. J. Vallerand (Ed.), *Les fondements de la psychologie sociale* (pp. 235‑291). Montréal: Gaétan Morin éditeur.

Ververidis, D., Kotropoulos, C. (2005). Emotional speech classification using Gaussian mixture models and sequential floating forward selection algorithm. *IEEE International Conference on Multimedia and Expo*, 1500–1503.

Xiao, Z., Dellandrea, E., Weibei Dou, Liming Chen. (2005). Features extraction and selection for emotional speech classification. *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance*. doi:10.1109/avss.2005.1577304

Wallon, H. (1934/2015). *Les origines du caractère chez l'enfant* (7e ed. ed.). Paris: P. U. F.

Wieser, I., Barros, P., Heinrich, S., Wermter, S. (2018). Understanding auditory representations of emotional expressions with neural networks. *Neural Computing and Applications*. doi:10.1007/s00521-018-3869-3

Williams, C., Stevens, K. (1981) Vocal correlates of emotional states. *Speech Evaluation in Psychiatry*. Grune and Stratton, 189–220.

Wu, C. H., Liang, W. B. (2011). Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information and semantic labels. *IEEE Transactions on Affective Computing*, 2, 10–21.

Yang, Z., Hirschberg, J. (2018) Predicting Arousal and Valence from Waveforms and Spectrograms Using Deep Neural Networks. *Proceedings Interspeech 2018*, 3092-3096, DOI: 10.21437/Interspeech.2018-2397.

You, M., Chen, C., Bu, J., Liu, J., Tao, J. (1997). Getting Started With Usa's : a speech undersimulated and actual stress database. *EUROSPEECH-97*, 4, 1743–1746.

Yu, C., Tian, Q., Cheng, F., Zhang, S. (2011). Speech Emotion Recognition Using Support Vector Machines. *Advanced Research on Computer Science and Information Engineering*, 215–220. doi:10.1007/978-3-642-21402-8_35

Zanna, M. P., Rempel, J. K. (1988). Attitudes: A new look at an old concept. D. Bar‑Tal & A. Kruglanski (Eds.), *The social psychology of knowledge* (pp. 315‑334). New York: Cambridge University Press.

Zhao, J., Mao, X., Chen, L. (2018). Learning deep features to recognise speech emotion using merged deep CNN. *IET Signal Processing*, 12, 713–721.

Zhang, Z., Wu, B., Schuller, B. (2019). Attention-augmented End-to-end Multi-task Learning for Emotion Prediction from Speech. *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6705–6709.

Zheng, W. Q., Yu, J. S., & Zou, Y. X. (2015). An experimental study of speech emotion recognition based on deep convolutional neural networks. *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. doi:10.1109/acii.2015.7344669

Annexe 1. Extraction des MFCC et création des images

```
import librosa
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
from pathlib import Path

#  fonction  pour  calculer  la  moyenne  de  la  dérivée  première  d'un
ensemble de valeurs ("delta")
def mean_delta(values, axis=0):
  delta = np.diff(values, axis=axis)
  return(np.mean(delta, axis=axis))

# fonction pour calculer la moyenne de la dérivée seconde d'un ensemble
de valeurs ("deltadelta")
def mean_deltadelta(values, axis=0):
  delta = np.diff(np.diff(values, axis=axis), axis=axis)
  return(np.mean(delta, axis=axis))

#  fonction  pour  exporter  un  tableau  2D  numpy  sous  forme  d'image  en
niveaux de gris
def  export_grayscale_image(numpy_data,  filename,  image_width_inches,
image_height_inches, resolution_dpi=None, vmin=None, vmax=None):
  fig = plt.figure(frameon=False)
  fig.set_size_inches(image_width_inches,image_height_inches)
  ax = plt.Axes(fig, [0., 0., 1., 1.])
  ax.set_axis_off()
  fig.add_axes(ax)
      ax.imshow(numpy_data,   aspect="auto",   cmap="gray",   vmin=vmin,
vmax=vmax)
  fig.savefig(filename, dpi = resolution_dpi)
  plt.close(fig)

############################

## definition des parametres
# dossiers contenant les extraits audio à analyser
```

```
dossiers_extraits_audio_par_classe = ('positif', 'neutre', 'negatif')
# calcul des MFCC
duree_max_signal_secondes = 2.5 # None pour utiliser la durée de
l'extrait le plus long présent dans les données
n_coefs_MFCC = 13
duree_trame_millisecondes = 25
recouvrement_entre_trames_millisecondes = 10
# export images
dossier_images = "imagesMFCC"
sousdossier_images_valeurs_brutes = "brut"
sousdossier_images_valeurs_normalisees = "norm"
sousdossier_images_valeurs_brutes_aleatoire = "brut_rnd"
sousdossier_images_valeurs_normalisees_aleatoire = "norm_rnd"
extension_images = ".png"
largeur_images_pouces = 7
hauteur_images_pouces = .9
resolution_images_dpi = 50
zero_padding_images = True # si False, les images plus petites
(extraits audios plus courts) sont étirées pour atteindre la même
largeur
zero_padding_blanc = True # si False, on complète avec des pixels noirs

##############################

## calcul des MFCC sur chaque extrait audio et stockage des valeurs
dans un tableau

dfRawValues = pd.DataFrame()
for targetdir in dossiers_extraits_audio_par_classe:
  for path in os.listdir(targetdir):
      print("%s - %s" % (targetdir, path))
      X, sample_rate = librosa.load(targetdir + '/' + path
                                    ,res_type='kaiser_fast'
                                    ,duration=duree_max_signal_secondes
                                              ,sr=None # frequence
d'echantillonnage determinee par librosa.load a partir de l'en-tete du
fichier .wav
                                    ,offset=0.0
                                   )
      sample_rate = np.array(sample_rate)
```

```
      # calcul des 13 coefs MFCC sur chaque trame
      mfccs_raw = np.transpose(librosa.feature.mfcc(y=X,
        sr=sample_rate,
        n_mfcc=n_coefs_MFCC,
               n_fft=int(sample_rate*duree_trame_millisecondes/1000),  #
extraction des MFCC sur des trames de duree_trame_millisecondes ms (cf.
https://github.com/librosa/librosa/issues/584)

hop_length=int(sample_rate*(duree_trame_millisecondes-recouvrement_entr
e_trames_millisecondes)/1000)))  # definition du decalage entre trames
consecutives (cf. https://github.com/librosa/librosa/issues/584)

       # stockage des informations dans le tableau dfRawValues avec les
infos sur la classe et le nom de l'extrait audio
      mfccs_raw_df = pd.DataFrame(mfccs_raw)
      mfccs_raw_df['classe'] = targetdir
      mfccs_raw_df['extrait'] = path
      dfRawValues = pd.concat([dfRawValues, mfccs_raw_df])

##############################

## normalisation des valeurs pour chacun des coefficients MFCC

# calcul de la moyenne et de l'ecart-type de chacun des coefficients
MFCC
npRawValues      =      dfRawValues.drop(columns      =      ['classe',
'extrait']).to_numpy()
npRawValuesColMeans = np.mean(npRawValues, axis = 0)
npRawValuesColStds = np.std(npRawValues, axis = 0)

# normalisation en z-scores de chacune des colonnes (soustraction de la
moyenne puis division par l'ecart-type)
npNormValues = (npRawValues - npRawValuesColMeans) / npRawValuesColStds

# on remet les valeurs normalisees dans une DataFrame avec les colonnes
classe et extrait
dfNormValues = pd.DataFrame(npNormValues)
dfNormValues['classe'] = dfRawValues['classe'].tolist()
dfNormValues['extrait'] = dfRawValues['extrait'].tolist()
```

```
##############################

## export dans des fichiers textes des valeurs brutes et normalisées
des MFCC pour chaque trame

dfRawValues.to_csv('valeurs_MFCC_brutes.txt', sep = '\t', index =
False)
dfNormValues.to_csv('valeurs_MFCC_norm.txt', sep = '\t', index = False)

##############################

## calcul des statistiques pour chaque extrait audio sur les valeurs
brutes et normalisées

## valeurs brutes
# calcul des statistiques sur chaque extrait pour chacun des
coefficients MFCC
statsRaw = dfRawValues.groupby(['classe', 'extrait']).agg(['mean',
'std', mean_delta, mean_deltadelta])
# conversion en colonnes des critères de regroupement classe et extrait
(par défaut indices de lignes)
statsRaw.reset_index(level=statsRaw.index.names, inplace=True)
# fusion des 2 niveaux hiérarchiques des noms de colonnes (indice des
MFCC et fonction d'aggrégation) en un seul niveau
# formatage conditionnel pour tenir compte des colonnes avec un seul
élément
statsRaw.columns = [f"{x}" if not y else f"{x}_{y}" for x, y in
statsRaw.columns.to_flat_index()]
# export du vecteur de paramètres pour chaque extrait
statsRaw.to_csv('stats_MFCC_brutes.txt', sep = '\t', index = False)

## valeurs normalisées
# calcul des statistiques sur chaque extrait pour chacun des
coefficients MFCC
statsNorm = dfNormValues.groupby(['classe', 'extrait']).agg(['mean',
'std', mean_delta, mean_deltadelta])
# conversion en colonnes des critères de regroupement classe et extrait
(par défaut indices de lignes)
statsNorm.reset_index(level=statsNorm.index.names, inplace=True)
```

```
# fusion des 2 niveaux hiérarchiques des noms de colonnes (indice des MFCC et fonction d'aggrégation) en un seul niveau
# formatage conditionnel pour tenir compte des colonnes avec un seul élément
statsNorm.columns = [f"{x}" if not y else f"{x}_{y}" for x, y in statsNorm.columns.to_flat_index()]
# export du vecteur de paramètres pour chaque extrait
statsNorm.to_csv('stats_MFCC_norm.txt', sep = '\t', index = False)

##############################

## export des images pour chacun des fichiers audio

# liste des extraits avec la classe correspondante
listeExtraits = dfRawValues[['classe','extrait']].drop_duplicates()

# valeurs minimum et maximum dans l'ensemble des données (version brute et normalisée) à prendre en compte lors du tracé des images
vmin_brut = np.amin(npRawValues)
vmax_brut = np.amax(npRawValues)
vmin_norm = np.amin(npNormValues)
vmax_norm = np.amax(npNormValues)
# valeur correspondant au blanc (ou noir) pour le zero-padding : minimum ou maximum dans l'ensemble des données
if zero_padding_blanc: # blanc = valeur max
  valeur_zero_padding_brut = vmax_brut
  valeur_zero_padding_norm = vmax_norm
else: # noir = valeur min
  valeur_zero_padding_brut = vmin_brut
  valeur_zero_padding_norm = vmin_norm

if zero_padding_images:
   # taille maximale en nombre de trames pour le "zero-padding" des images plus petites
               nMaxTrames        =        dfRawValues.groupby(['classe', 'extrait'])[0].count().max()

# création des dossiers de destination si nécessaire
Path(dossier_images).mkdir(parents=True, exist_ok=True)
```

```
# un sous-dossier par type d'image (données brutes ou normalisées, avec
ou sans ordre aléatoire)
target_subfolders          =          [sousdossier_images_valeurs_brutes,
sousdossier_images_valeurs_normalisees,
sousdossier_images_valeurs_brutes_aleatoire,
sousdossier_images_valeurs_normalisees_aleatoire]
classes = listeExtraits['classe'].unique()
for subfolder in target_subfolders:
  Path(dossier_images+'/'+subfolder).mkdir(parents=True, exist_ok=True)
  # un sous-dossier par classe dans chacun de ces sous-dossiers
  for classe in classes:
       Path(dossier_images+'/'+subfolder+'/'+classe).mkdir(parents=True,
exist_ok=True)
for index, row in listeExtraits.iterrows():
  classe = row['classe']
  extrait = row['extrait']
  # traitement des valeurs brutes
                                    npRawExtraitCourant                 =
dfRawValues[dfRawValues['extrait']==extrait].drop(columns  =  ['classe',
'extrait']).to_numpy()
  # version avec les trames mélangées en ordre aléatoire
  npRawExtraitCourantRnd = np.copy(npRawExtraitCourant)
  np.random.shuffle(npRawExtraitCourantRnd)

   # ajout de zéros a la fin si le nombre de lignes est inférieur au
maximum ("zero padding")
  if zero_padding_images and npRawExtraitCourant.shape[0]<nMaxTrames:
              npRawExtraitCourant    =    np.append(npRawExtraitCourant,
np.ones((nMaxTrames-npRawExtraitCourant.shape[0],
13))*valeur_zero_padding_brut, axis=0)
          npRawExtraitCourantRnd   =   np.append(npRawExtraitCourantRnd,
np.ones((nMaxTrames-npRawExtraitCourant.shape[0],
13))*valeur_zero_padding_brut, axis=0)

  # traitement des valeurs normalisées
                                    npNormExtraitCourant                =
dfNormValues[dfNormValues['extrait']==extrait].drop(columns            =
['classe', 'extrait']).to_numpy()
  # version avec les trames mélangées en ordre aléatoire
  npNormExtraitCourantRnd = np.copy(npNormExtraitCourant)
```

```
  np.random.shuffle(npNormExtraitCourantRnd)

   # ajout de zéros a la fin si le nombre de lignes est inférieur au
maximum ("zero padding")
  if zero_padding_images and npNormExtraitCourant.shape[0]<nMaxTrames:
            npNormExtraitCourant    =    np.append(npNormExtraitCourant,
np.ones((nMaxTrames-npNormExtraitCourant.shape[0],
13))*valeur_zero_padding_norm, axis=0)
         npNormExtraitCourantRnd   =   np.append(npNormExtraitCourantRnd,
np.ones((nMaxTrames-npNormExtraitCourant.shape[0],
13))*valeur_zero_padding_norm, axis=0)

  # construction et export des images dans les sous-dossiers cibles
               export_grayscale_image(npRawExtraitCourant.transpose(),
dossier_images+"/"+sousdossier_images_valeurs_brutes+"/"+classe+"/"+ext
rait+extension_images,   largeur_images_pouces,   hauteur_images_pouces,
resolution_images_dpi, vmin_brut, vmax_brut)
             export_grayscale_image(npRawExtraitCourantRnd.transpose(),
dossier_images+"/"+sousdossier_images_valeurs_brutes_aleatoire+"/"+clas
se+"/"+extrait+extension_images,                largeur_images_pouces,
hauteur_images_pouces, resolution_images_dpi, vmin_brut, vmax_brut)
               export_grayscale_image(npNormExtraitCourant.transpose(),
dossier_images+"/"+sousdossier_images_valeurs_normalisees+"/"+classe+"/
"+extrait+extension_images,                     largeur_images_pouces,
hauteur_images_pouces, resolution_images_dpi, vmin_norm, vmax_norm)
            export_grayscale_image(npNormExtraitCourantRnd.transpose(),
dossier_images+"/"+sousdossier_images_valeurs_normalisees_aleatoire+"/"
+classe+"/"+extrait+extension_images,           largeur_images_pouces,
hauteur_images_pouces, resolution_images_dpi, vmin_norm, vmax_norm)
```

Annexe 2. Concaténation des valeurs

```
# -*- coding: utf-8 -*-

import os
import pandas as pd

df = pd.read_csv("/valeurs_MFCC_norm.txt", sep="\t")
# les valeurs sont regroupés par extrait et par classe
df_concat                                                    =
df.set_index(['extrait','classe',df.groupby(['extrait','classe']).cumco
unt()+1]).unstack().sort_index(level=1, axis=1)
#     les     noms     de     colonnes     prennent     le     format
*pas-de-25ms*_*numéro-de-coefficient*
df_concat.columns = df_concat.columns.map('{0[1]}_{0[0]}'.format)
# df_concat

# dans dataframe créé les coefficients ne sont pas dans l'ordre correct
# les coefficients 10, 11, 12 viennent avant le coefficient 2
# on extrait les noms de colonnes
colonnes = df_concat.columns.tolist()
print(colonnes)

#on met les noms des colonnes dans l'ordre correct
column_names = ['1_0', '1_1', '1_2', '1_3', '1_4', '1_5', '1_6', '1_7',
'1_8', '1_9', '1_10', '1_11', '1_12',
 '2_0', '2_1', '2_2', '2_3', '2_4', '2_5', '2_6', '2_7', '2_8', '2_9',
'2_10', '2_11', '2_12',
 '3_0', '3_1', '3_2', '3_3', '3_4', '3_5', '3_6', '3_7', '3_8', '3_9',
'3_10', '3_11', '3_12',
                        . . . . . .
  '166_0',  '166_1',  '166_2',  '166_3',  '166_4',  '166_5',  '166_6',
'166_7', '166_8', '166_9', '166_10', '166_11', '166_12',
  '167_0',  '167_1',  '167_2',  '167_3',  '167_4',  '167_5',  '167_6',
'167_7', '167_8', '167_9', '167_10', '167_11', '167_12']

# les noms des colonnes sont reindexés
df_reindex = df_concat.reindex(columns=column_names)
```

```
df_reindex

# extraction de la valeur maximale des données
max_values = max(list(df_reindex.max()))
# print(max_values)
# remplacement des valeurs manquantes par la valeur maximale
df_zero_padding = df_reindex.fillna(max_values)
# df_zero_padding[20:40]

df_zero_padding.to_csv("/norm_MFCC_concat.txt")
```

.

Annexe 3. Création du SVM

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
import seaborn as sns

#on prend les données issus de l'annexe 1 pour les stats et issus de
l'annexe 2 pour les valeurs MFCC
data = pd.read_csv('/norm_MFCC_concat.txt')

# print(data.shape)
# print(data.head())
#on sépare les valeurs dans la variable X et les étiquettes dans la
variable y
X = data.drop(['classe', 'extrait'], axis=1)
y = data["classe"]

# print(X.head())
# on sépare au hasard les données en test et train
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.25, random_state=42)

#on applique l'algorithme de la classification du svm
# la classification binaire se réalise par l'approche one-vs-one, par
défaut c'est one-vs-all
# la fonction de kernel est noyau polynomial
svclassifier = SVC(kernel='poly', decision_function_shape='ovo')
svclassifier.fit(X_train, y_train)

# teste du modèle
y_pred = svclassifier.predict(X_test)

# print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
def   joli_CM(confusion_matrix,   noms_classes,   figsize   =   (10,7),
fontsize=14):
  df_cm = pd.DataFrame(
        confusion_matrix, index=noms_classes, columns=noms_classes,
    )
  fig = plt.figure(figsize=figsize)
  heatmap = sns.heatmap(df_cm, annot=True, fmt="d")

           heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(),
rotation=0, ha='right', fontsize=fontsize)
           heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(),
rotation=45, ha='right', fontsize=fontsize)
  plt.ylabel('True label')
  plt.xlabel('Predicted label')

classes = ("negatif", "neutre", "positif")
cm = confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test, y_pred))
joli_CM(cm, noms_classes = classes)
```

Annexe 4. Création du CNN à une dimension

```
import os
import numpy as np
import librosa
import pandas as pd
import keras
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout, Activation,
BatchNormalization
from keras.layers import Conv1D, MaxPooling1D
from keras.utils import np_utils, to_categorical
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

###############CONSTRUSTION DU MODELE############################
#séparation au hasarddes donnés en ensemble de test et de train
# les variable X_train et X_test contiennent des valeurs et leurs
indexes qui seront données pour l'apprentissage et pour le test au
modèle
# y_train et y_test contiennent les étiquettes et leur indexes
dataset = pd.read_csv('/norm_MFCC_concat.txt')
X_train,          X_test,          y_train,          y_test          =
train_test_split(dataset.drop(['classe','extrait'],axis=1)
                                                    , dataset.classe
                                                    , test_size=0.25
                                                    , shuffle=True
                                                    , random_state=42
                                                   )


# transformation des objets dataframes et series issus de l'étape de la
séparation des données au format array de numpy
X_train = np.array(X_train)
y_train = np.array(y_train)
X_test = np.array(X_test)
y_test = np.array(y_test)
```

```
# labEn.fit_transform permet de transformer l'étiquette de l'extrait
sous une forme numérique : par exemple, neutre devient 0
# np_utils.to_categorical met les données sous la forme de one hot
vector : 0 devient 1, 0, 0
# transformation complète est la suivante : neutre -> 0 -> 0, 0, 1
labEn = LabelEncoder()
y_train = np_utils.to_categorical(labEn.fit_transform(y_train))
y_test = np_utils.to_categorical(labEn.fit_transform(y_test))
# np.expand_dims fait la matrice des valeurs
X_train = np.expand_dims(X_train, axis=2)
X_test = np.expand_dims(X_test, axis=2)

# construction du modèle
model = Sequential()
model.add(Conv1D(256,                                                    8,
padding='same',input_shape=(X_train.shape[1],1)))
model.add(Activation('relu'))
model.add(Conv1D(256, 8, padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))
model.add(MaxPooling1D(pool_size=(8)))
model.add(Conv1D(128, 8, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(128, 8, padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))
model.add(MaxPooling1D(pool_size=(6)))
model.add(Conv1D(64, 8, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(64, 8, padding='same'))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(3)) # nombre de classes
model.add(Activation('softmax'))
opt = keras.optimizers.RMSprop(lr=0.00001, decay=1e-6)
model.compile(loss='categorical_crossentropy',      optimizer='RMSprop',
metrics=['accuracy'])
model.summary()
```

```
# test du modèle
model.compile(loss='categorical_crossentropy',      optimizer='RMSprop',
metrics=['accuracy'])
model_history=model.fit(X_train,   y_train,   batch_size=32,   epochs=15,
validation_data=(X_test, y_test))

preds = model.predict(X_test,batch_size=32,verbose=1)
preds = preds.argmax(axis=1)
print(preds)

# étiquettes par le modèle
preds = preds.astype(int).flatten()
preds = (labEn.inverse_transform((preds)))
preds = pd.DataFrame({'predictedvalues': preds})
# étiquettes réelles
actual = y_test.argmax(axis=1)
actual = actual.astype(int).flatten()
actual = (labEn.inverse_transform((actual)))
actual = pd.DataFrame({'actualvalues': actual})

print(classification_report(actual, preds,
                            target_names=np.unique(y_test))
```

Annexe 5. Séparation des images en ensembles d'apprentissage et de test

```
import os
import numpy as np
import shutil
import random
# création des noms des dossiers
target_dos = 'norm_rnd'
classes_dos = ['/negatif', '/positif', '/neutre']
# on établit la taille du test set
test_ratio = 0.25
# création des dossiers pour le test et train sets
for cls in classes_dos:
    os.makedirs(target_dos +'/train' + cls)
    os.makedirs(target_dos +'/test' + cls)
# nom du dossier duquel on copie des images
    source = target_dos + cls
# on récupère les noms de tous les fichiers dans le dossier pour chaque
polarité
    noms_fics = os.listdir(source)
# les données sont randomisées
    np.random.shuffle(noms_fics)
# les données sont divisiées
      noms_fics_train, noms_fics_test = np.split(np.array(noms_fics),
[int(len(noms_fics)* (1 - test_ratio))])
            noms_fics_train   =   [source+'/'+   nom   for   nom   in
noms_fics_train.tolist()]
           noms_fics_test   =   [source+'/'   +   nom   for   nom   in
noms_fics_test.tolist()]
    print('nombre total des images dans un dossier de la polarité : ',
len(noms_fics))
    print('nombre de fichiers dans train set : ', len(noms_fics_train))
   print('nombre de fichiers dans un test set : ', len(noms_fics_test))
# on copie des images
    for nom in noms_fics_train:
        shutil.copy(nom, target_dos +'/train' + cls)
    for nom in noms_fics_test:
        shutil.copy(nom, target_dos +'/test' + cls)
```

Annexe 6. Création du CNN à deux dimensions

```
from google.colab import files, drive
drive.mount('/content/gdrive/')
#importation des librairies
import os
import cv2
import numpy as np
from glob import glob as glb
from keras import layers
from keras import models
from keras.models import Sequential
from keras.utils import np_utils
import random
import pandas as pd
from keras.layers import Input, Flatten, Dropout, Activation,
BatchNormalization, ZeroPadding2D, MaxPooling2D, Dense
from keras.layers import Convolution2D as Conv2D
from sklearn.metrics import accuracy_score, classification_report

# création du dossier pour dézipper les données
!mkdir corpus_norm

!unzip '/content/gdrive/My Drive/norm_mfcc.zip' -d "/content/gdrive/My
Drive/corpus_norm"

# récupération des noms des dossiers = noms des classes
noms_classes = [l.split('/')[-1] for l in glb('/content/gdrive/My
Drive/corpus_norm/*/*')]
# on assigne à chaque classe un chiffre
dico_classes = {'positif':0, 'neutre':1, 'negatif':2}
print(noms_classes)
print(dico_classes)

# création des listes des données et des étiquettes d'entrainement et
de test
X_train = []
y_train = []
```

```
X_test = []
y_test = []

# pour chaque dossier des classes
for classe in noms_classes:
# pour chaque fichier dans le dossier
                for         chemin         in         glb('/content/gdrive/My
Drive/corpus_norm/*/{}/train/*'.format(classe)):
# load l'image aux nuances de gris au format uint8
    x = cv2.imread(chemin, 0).astype(np.uint8)
# on remet tous les images au même format
# une plus grande taille des images ralentit le modèle et ne donne pas
de meilleurs résultats
    x = cv2.resize(x, (28, 28))
    y = dico_classes[classe]
    X_train.append(x)
    y_train.append(y)

                for         chemin         in         glb('/content/gdrive/My
Drive/corpus_norm/*/{}/test/*'.format(classe)):
    x = cv2.imread(chemin, 0).astype(np.uint8)
    x = cv2.resize(x, (28, 28))
    y = dico_classes[classe]
    X_test.append(x)
    y_test.append(y)

print(len(y_test))
print(len(y_train))

# les données sont mises au format de numpy
XX_train = np.array(X_train)
# elles sont adaptées pour le CNN à deux dimensions
XX_train  =  XX_train.reshape(-1,  XX_train.shape[1],  XX_train.shape[2],
1)
#  normalisation  des  images  -  division  par  la  plus  grande  valeur  pour
l'image type
XX_train = XX_train.astype(np.uint8) / 255

XX_test = np.array(X_test)
XX_test = XX_test.reshape(-1, XX_test.shape[1], XX_test.shape[2], 1)
```

```
XX_test = XX_test.astype(np.uint8) / 255

# mettres les étiquettes sous l'encodage de one-hot
yy_train = np_utils.to_categorical(y_train, len(noms_classes))
yy_test = np_utils.to_categorical(y_test, len(noms_classes))

#Modèle
model = Sequential()
#ajoute les filtres: 32 couches de la taille 5x5 en mode relu et la
taille d'entrée de l'image est de 28, 28
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=(28,28,1)))
model.add(Conv2D(64, (3, 3), activation='relu'))
# couche de pooling avec le filtre de la taille 2*2
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(3, activation='softmax'))
model.summary()

#teste du modèle
model.compile(loss='categorical_crossentropy',optimizer='RMSprop',metri
cs = ['accuracy'])
history = model.fit(XX_train, yy_train,
                  validation_data=(XX_test, yy_test),
                  batch_size=32,
                  epochs=15,
                  verbose=1)

# récupération des étiquettes après le test
predictions = model.predict(XX_test,
                      batch_size=32,
                      verbose=1)
predictions=predictions.argmax(axis=1)

# mettre les prédictions sous l'encodage one-hot
preds_array = np.array(predictions)
```

```
preds_one_hot = np.zeros((preds_array.size, preds_array.max()+1))
preds_one_hot[np.arange(preds_array.size), preds_array] = 1

print(classification_report(yy_test, preds_one_hot))
```