|  |  |  |
| --- | --- | --- |
| tri_logo | Université Chouaib Doukkali  Ecole Nationale des Sciences Appliquées d’El Jadida  Département Télécommunications, Réseaux et Informatique |  |

Mise en place d’une application de collecte de corpus audio depuis le web pour l’identification automatique des langues : Cas des dialectes marocains

**Projet de Fin d’Année – [PFA]**

**IITE**

##### **Réalisé par :**

Abdelhamid BENSITEL - Mustapha EDDAMNNI

##### **Encadré par :**

Pr. A.EL HANNANI

**El Jadida**

Année universitaire 2019/2020

Résumé

Récemment, les tentatives dans le domaine du traitement automatique de la parole TAP pour la langue arabe ont éveillé l’attention de quelques chercheurs. En fait, la majorité de ces tentatives ont mis l’accent sur la norme officielle de la langue arabe qui est connu comme l’arabe moderne standard. En revanche, l’arabe ne présente pas la langue des communications courantes dans tous les pays arabes. De ce fait, on conçoit l’existence de plusieurs variétés arabes utilisées dans la vie quotidienne pour la communication ordinaire des communautés. Certes, ces différents dialectes arabes possèdent une forme parlée et non écrite et se distinguent par des caractéristiques phonologiques, morphologiques, syntaxiques et lexicales importantes qui se diffèrent d’un dialecte à un autre.

Dans la plupart des langues peu dotées, les services liés aux technologies du traitement de l’oral sont inexistants. L’originalité de notre projet vient de la volonté d’aborder ces langues pour lesquelles peu ou pas de ressources nécessaires pour la reconnaissance automatique de la parole sont disponibles. Ce manuscrit présente notre méthodologie qui vise à développer et adapter rapidement un système de reconnaissance automatique de la parole continue pour une nouvelle langue peu dotée. Le dialecte marocain, par exemple est un important dialecte véhiculaire du Maroc. En ce qui concerne la structure de la langue elle se distingue par son vocabulaire d’origine arabe d’un côté et amazigh d’un autre. Pour l’implémentation d’un tel système de la reconnaissance automatique de la parole avec une nouvelle langue (Les dialectes marocains comme exemple), nous avons besoin de collecter des corpus de la parole de cette langue ce qui nous facilitera par la suite son identification d’une manière automatique.

Le premier chapitre de ce projet comporte une brève définition du traitement automatique de la parole TAP. Ainsi, ce rapport présente les différentes étapes de la collecte d’un corpus audio depuis le web. Pour qui suit, nous détaillerons les étapes de l’extraction des audio, leur traitement et leur évaluation. Pour finir, nous exposons la validation des résultats de notre projet.

**Mots clés :** Traitement Automatique de la Parole, TAP, données audio, Identification automatique de la parole, la recherche des vidéos, corpus audio, Identification automatique du dialecte.

Remerciements

Au début de ce rapport, nous exprimons nos remerciements à toutes les personnes qui ont nous soutenue à accomplir notre projet de fin d’année et à le réaliser dans les meilleures conditions.

Tout d’abord, nous remercions les membres de jury Madame Asmaa EL HANNANI et Monsieur Mohamed LACHGAR qui ont fait le grand honneur de juger ce travail.

Nous adressons, aussi, nos vifs remerciements à notre encadrante Madame Asmaa EL HANNANI, pour son aide précieuse dans le travail, sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion.

Nous tenons à remercier aussi Monsieur Rahhal ERRATTAHI qui nous a donné une idée complète sur le déroulement du travail, pour l’interprétation des résultats des différentes tâches. Sa compétence et sa disponibilité ont toujours été une source d’encouragement.

Nos remerciements s’adressent également à l’ensemble du corps enseignant de l’ENSA El-Jadida, pour le temps passé à nous expliquer les bases dont un ingénieur en a besoin.

Nous tenons enfin à saluer nos parents et les remercier pour leur soutien actuel ou futur dans la vie et pour nous avoir poussés à réaliser ce projet.



Table des matières

[1 Introduction 8](#_Toc513981328)

[2 Généralité et Etat de l’Art 10](#_Toc513981329)

[2.1 Les dialectes marocains 10](#_Toc513981330)

[2.1.1 L’Amazigh 15](#_Toc513981333)

[2.1.2 L’Arabe dialectal 15](#_Toc513981333)

[2.2 Deduction 10](#_Toc513981330)

[2.2.1 L’Arabe dialectal 15](#_Toc513981333)

[2.2.2 L’Amazigh 15](#_Toc513981333)

[2.3 Etat de l’art des corpus en dialecte 15](#_Toc513981332)

[2.3.1 Corpus existants 17](#_Toc513981334)

[2.4 Exemple d’un site de la reconnaisance vocale 19](#_Toc513981335)

[2.5 Outils de travail 22](#_Toc513981335)

[3 La collecte d’un corpus audio depuis le web pour les dialectes marocains](#_Toc513981345)

[3.1 Introduction 24](#_Toc513981346)

[3.2 L’inventaire des sources vidéo 24](#_Toc513981346)

[3.3 Les étapes de la construction du corpus audio 25](#_Toc513981347)

[3.3.1 La recherche des chaines IDs 25](#_Toc513981348)

[3.3.2 L’extraction des IDs vidéos 26](#_Toc513981349)

[3.3.3 L’extraction des informations des audios 27](#_Toc513981350)

[3.3.4 Filrage des audios 27](#_Toc513981351)

[3.3.5 L’extraction des audios (crawl) 28](#_Toc513981351)

[3.3.6 La segmentation des audios 28](#_Toc513981351)

[3.3.7 La segmentation physique des audios 29](#_Toc513981351)

[3.4 Evaluation des résultats 29](#_Toc513981347)

[4 Validation des résultats 36](#_Toc513981345)

[5 Conclusion et Perspectives 37](#_Toc513981353)

[Bibliographie 38](#_Toc513981354)

Liste des figures

[Figure 3 : Structure hiérarchique des dialectes marocains. 10](#_Toc513980780)

[Figure 4 : Les régions dont le parler est l'arabe. 11](#_Toc513980781)

[Figure 5: Les régions dont le parler est l'amazigh. 12](#_Toc513980782)

[Figure 6 : Exemple de corpus de parole Européens et Anglais . 17](#_Toc513980782)

[Figure 7 : Caractéristiques des corpus d’arabes dialectaux étudiés 18](#_Toc513980783)

[Figure 8 : Avant la capture de l’audio. 20](#_Toc513980784)

[Figure 9 : Après la capture de l’audio. 21](#_Toc513980785)

[Figure 10 : Logo YouTube Data API . 18](#_Toc513980785)

[Figure 11 : Logo TensorFlow. 18](#_Toc513980785)

[Figure 12 : Logo Pydub. 19](#_Toc513980785)

[Figure 13 : Logo FFmpeg. 19](#_Toc513980785)

[Figure 14 : Exemple chaines IDs. 21](#_Toc513980786)

[Figure 15 : Vidéos IDs ,dialecte label ,genre et durée 22](#_Toc513980787)

[Figure 16 : Exemple du fichier segments. 24](#_Toc513980788)

[Figure 17 : Exemple de segmentation physique 24](#_Toc513980789)

[Figure 18 : Graphe statistiques de parole 27](#_Toc513980790)

[Figure 19 : Graphe statistiques de segments 28](#_Toc513980792)

[Figure 20 : Graphe statistiques durée des segments 28](#_Toc513980793)

Liste des tableaux

[Tableau 1 : Applications convertisseur. 13](#_Toc513975328)

[Tableau 2 : Nombre d’heures initial par dialecte. 25](#_Toc513975329)

[Tableau 3 : Statistiques de parole pour chaque dialecte 26](#_Toc513975330)

[Tableau 4 : Statistiques de segments pour chaque dialecte 27](#_Toc513975330)

[Tableau 5 : Classification de segments pour chaque dialecte 28](#_Toc513975330)

[Tableau 6 : Totale d’heures des audios supprimés et obtenues 29](#_Toc513975337)

1. Introduction

Au cours des dernières années, l’exploration du Web et l’enregistrement des émissions TV sont devenus les principales sources de données pour la recherche et le progrès dans le domaine du Traitement Automatique de la Parole (TAP) qui est un domaine multidisciplinaire impliquant la [linguistique](https://fr.wikipedia.org/wiki/Linguistique), l'[informatique](https://fr.wikipedia.org/wiki/Informatique) et l'[intelligence artificielle](https://fr.wikipedia.org/wiki/Intelligence_artificielle), qui vise à créer des outils de traitement de la langue naturelle pour diverses applications. Il ne doit pas être confondu avec la [linguistique informatique](https://fr.wikipedia.org/wiki/Linguistique_informatique), qui vise à comprendre les langues au moyen d'outils informatiques.

Le but de ce projet est de développer une application pour la recherche des vidéos puis l’extraction des données audio en utilisant l’API « YouTube Data API », afin de constituer un corpus audio du dialecte de l’arabe marocaine depuis le web. Les données collectées doivent être évaluées par la suite en utilisant un système d’identification automatique du dialecte et de proposer des approches visant à augmenter la robustesse des systèmes de TAP face au dialecte marocain.

La nature statistique des approches nécessite de disposer d’une grande quantité de ressources (grands corpus de parole) pour le développement d’un système d’identification automatique de la parole. Ces ressources ne sont cependant pas disponibles directement pour des dialectes de l’arabe marocaine. Par conséquent, une première façon d’accélérer la portabilité des systèmes d’identification de paroles vers une nouvelle langue est de développer une méthodologie permettant une collecte rapide de ressources orales.

Dans le cas idéal du traitement automatique de la parole on vise la recherche des vidéos puis l’extraction les données audios en utilisant l’API afin d’avoir un large corpus de parole pour la langue maternelle parlée « Darija ».

Cependant, l'acquisition de tels corpus serait une opération coûteuse et difficile à réaliser, vu qu'il existe un très grand nombre de langues parlées dans l’arabe marocaine « Hassania, Chelha, Rifia … ».

1. Généralité et Etat de l’Art
   1. Les dialectes marocains

Le [Maroc](https://fr.wikipedia.org/wiki/Maroc) compte deux langues officielles : **l'**[**amazigh**](https://fr.wikipedia.org/wiki/Langues_berb%C3%A8res) et **l'arabe dialectal**. Chacune des deux langues est parlée sous différentes formes dialectales, alors que l'[arabe littéral](https://fr.wikipedia.org/wiki/Arabe_litt%C3%A9ral) est la langue administrative et que l'amazighe ne possède pas de forme unifiée ; cependant une [version standardisée de l'amazighe](https://fr.wikipedia.org/wiki/Amazighe_standard_marocain) est progressivement introduite.

* + 1. L’amazigh

D’après plusieurs historiens, les Amazighes sont les premiers habitants du Maroc. Seulement, ils ont été arabisés dans les plaines alors que dans les montagnes-refuges du Maroc, ils ont su protéger jusqu’à nos jours leur langue et leurs coutume.

Ainsi, la dénomination berbère (dérivé de barbare), désuète et rejetée à cause du sens péjoratif, a été remplacée par amazighe ou tamazight dans la littérature linguistique et culturelle maghrébine, bien que dans les faits, ces termes restent synonymes. Par ailleurs, les natifs de cette langue se désignent eux-mêmes par le terme amazigh, ce qui signifie « homme noble » ou « homme libre ». La terminologie officielle du gouvernement marocain utilise aussi le terme amazighe (ou amazigh).

Parmi les évènements qui ont mis cette langue en valeur, nous citons notamment : la création de l’institut royal de la culture amazighe (L’IRCAM) le 17 octobre 2001 [2], l’introduction de cette langue par les mass médias (Radio et télévision diffusent informations, documentaires et films en amazigh).

* + 1. L’arabe dialectal

L’arabe dialectal (AD) ou l’arabe marocain (AM) dit « darija » est la langue maternelle des marocains non amazighophones. Ce dialecte dérive de l’arabe classique (AC) qui, à travers la succession des générations, va subir des transformations qui lui assureront une certaine autonomie. Moatassim (1974 : 642) [1] confirme que : « apparenté de l’arabe classique, l’arabe dialectal est une variante au vrai sens du mot. Mais comme l’amazighe, il reste la langue de l’authenticité pour beaucoup de Marocains. »

Malgré ses variétés, l’AD, langue parlée à la maison comme dans la rue, unifie les différentes communautés, assure l’intercompréhension en adoptant l’appellation générique « arabe marocain » (AM). Il est étendu sur tout le territoire marocain, mais appartient exclusivement à l’oralité, ce qui n’a pas empêché certains auteurs de l’utiliser récemment dans leurs productions littéraires.

Par ailleurs, cette langue n’a jamais été standardisée et donc, n’a aucun statut constitutionnel, à l’inverse de l’arabe classique, totalement absente de la vie quotidienne, et elle est la langue officielle du pays.

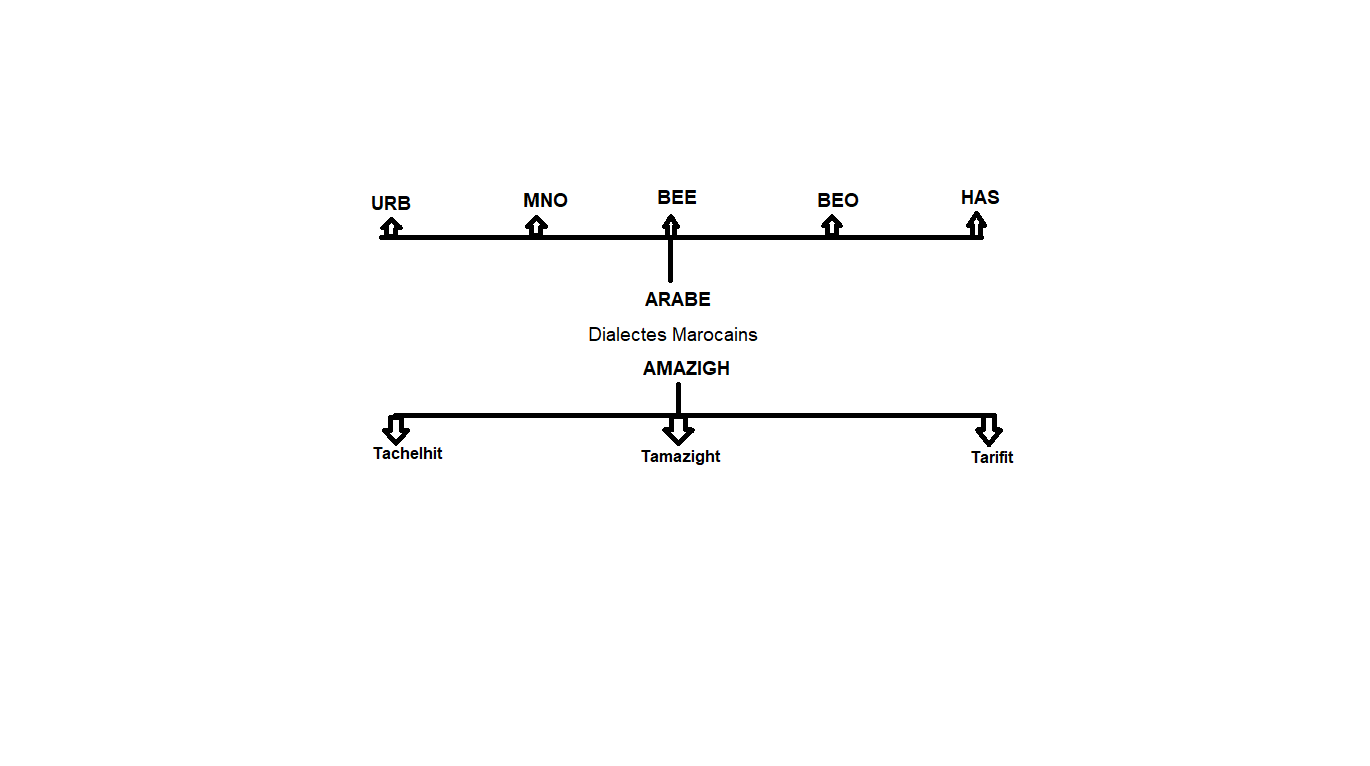
****

Figure 3 : Structure hiérarchique des dialectes marocains.

2.2 Déduction

Après toutes les informations déduites des ressources mentionnées ci-dessus, les langues sur lesquelles on va se baser sont l’Arabe dialectal et l’Amazigh.

Plus précisément, (l’Urbain, l’Arabe des montagnes du Nord-Ouest, le Bédouin de l’Ouest ainsi que celui de l’Est et la Hassanya) pour l’Arabe.

Et pour l’Amazigh, on se basera sur (le Tachelhit et le Tarifit).

2.2.1 L’Arabe dialectal

* L’**Urbains,** employé dans les villes marocaines traditionnelles telles que Fès, Rabat, Salé et Tétouan et marqué par des caractéristiques des parlers andalous.
* Le parler des **Montagnes** du **Nord-Ouest**, les environ de Tanger.
* Le **Bedouin** **de l’Ouest** employé par la communauté arabophone d'origine de Baní Hilal et de Baní Maaqil habitant les plaines de Gharb, de Chawiyya, de Doukkala et des villes voisines telles que Mohammedia, Casablanca, El Jadida, Marrakech, etc.
* Le **Bedouin de l’est** dit bedoui est le parler des gens qui habitent les environs d’Oujda, Berkane, Benidrar,…
* **Hassanya**, employée par des locuteurs d'origine de Baní Maaqil vivant maintenant au Sahara dans la région méridionale du Maroc.

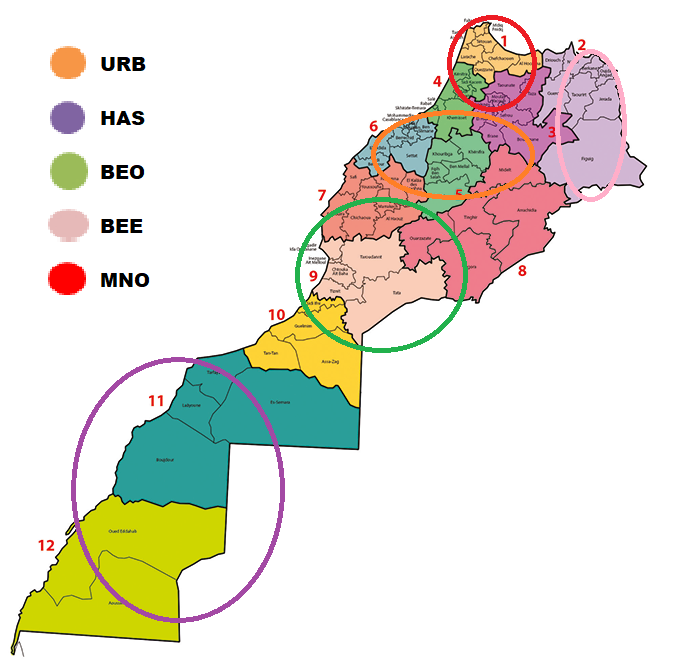


Figure 4 : Les régions dont le parler est l'arabe.

2.2.2 L’Amazigh

Aujourd’hui, l’amazigh est considéré comme un terme générique, lorsqu’il s’agit du « berbère standardisé », désignant plusieurs parlers classés par les chercheurs en trois variétés : le tarifit, le tamazight et le tachelhit.

* Le **Rifain** est utilisé au Nord, de Tanger à Nador en escaladant la chaîne du Rif.
* Le **Tachelhit** est parlé dans le Sud-Ouest d'Essaouira à Sidi Ifni en passant à travers l'Atlas Occidental, par le col de n'Tichka. Il est la langue des [chleuhs](http://fr.wikipedia.org/wiki/Chleuhs)
* Le **Tamazight** est le plus répandu, comprenant tout le reste du Maroc jusqu'aux portes du désert.

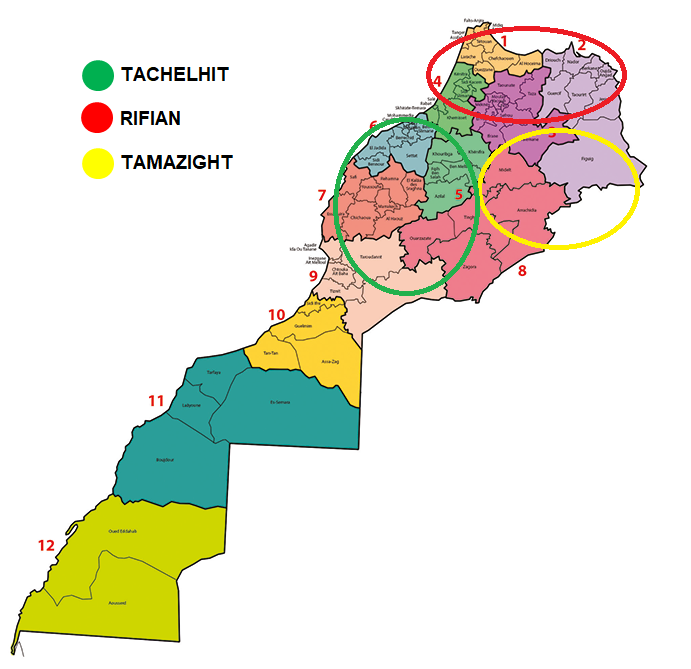


Figure 5 : Les régions dont le parler est l'amazigh.

En ce qui concerne Tamazight et Tachelhit, la recherche sur YouTube ne nous aide pas à différencier et à mieux distinguer entre les deux. Pour ceci, on a limité ces deux dialectes en Tachelhit.

* 1. Etat de l’art des corpus en dialecte

2.3.1 Corpus existants

1. Exemple de corpus audios qui traitent la reconnaissance automatique pour les dialectes des pays européens et Anglais [4] :

La production de corpus audios pour la reconnaissance automatique de la parole s’est faite à des degrés divers dans les différents pays travaillant sur le TAP (France, Etats-Unis, Angleterre, Allemagne, Pays-Bas, Italie, Espagne, Grèce… et plus récemment les pays asiatiques). Pour les langues européennes cette production a souvent été soutenue par des projets européens et trans-nationaux (corpus CGN pour les néerlandophones de Belgique et des Pays-Bas). Au niveau national une partie importante de la production de corpus est soutenue par la défense (ARPA (Advanced Research Project Agency) aux Etats-Unis, DGA (Délégation Générale à l’Armement) en France).

La figure 7 [3] présente certaines caractéristiques des corpus Européens et Anglais.

Le premier corpus, s’appelle TIDIGITS [16] Ce corpus contient de la parole qui a été initialement conçue et collectée par Texas Instruments, Inc. (TI) dans le but de concevoir et d'évaluer des algorithmes pour la reconnaissance indépendante du locuteur des séquences de chiffres connectés. Il y a 326 locuteurs (111 hommes, 114 femmes, 50 garçons et 51 filles) prononçant chacun des séquences de 77 chiffres.

EUROM [17] Le corpus EUROM.1 pour le portugais européen a été collecté dans le cadre du projet européen SAM\_A, conjointement par l'INESC et le CLUL. Ce projet était en fait le prolongement d'un projet préliminaire (SAM - Speech Assessment Methods) au cours duquel un travail sur la planification d'une ressource poly-langue pour les besoins en ingénierie linguistique parlée de l'Union européenne a commencé.

TIMIT [18] Le corpus TIMIT de la parole lue est conçu pour fournir des données vocales pour les études acousto-phonétiques et pour le développement et l'évaluation de systèmes de reconnaissance automatique de la parole. TIMIT contient des enregistrements à large bande de 630 locuteurs de huit principaux dialectes de l'anglais américain, chacun lisant dix phrases phonétiquement riches.

BDSONS [19] Le corpus se compose de 32 haut-parleurs: 16 hommes et 16 femmes (7 CD-ROM d'environ 3,5 gigaoctets), étiquetage phonétique (en partie) disponible sur des disquettes supplémentaires,

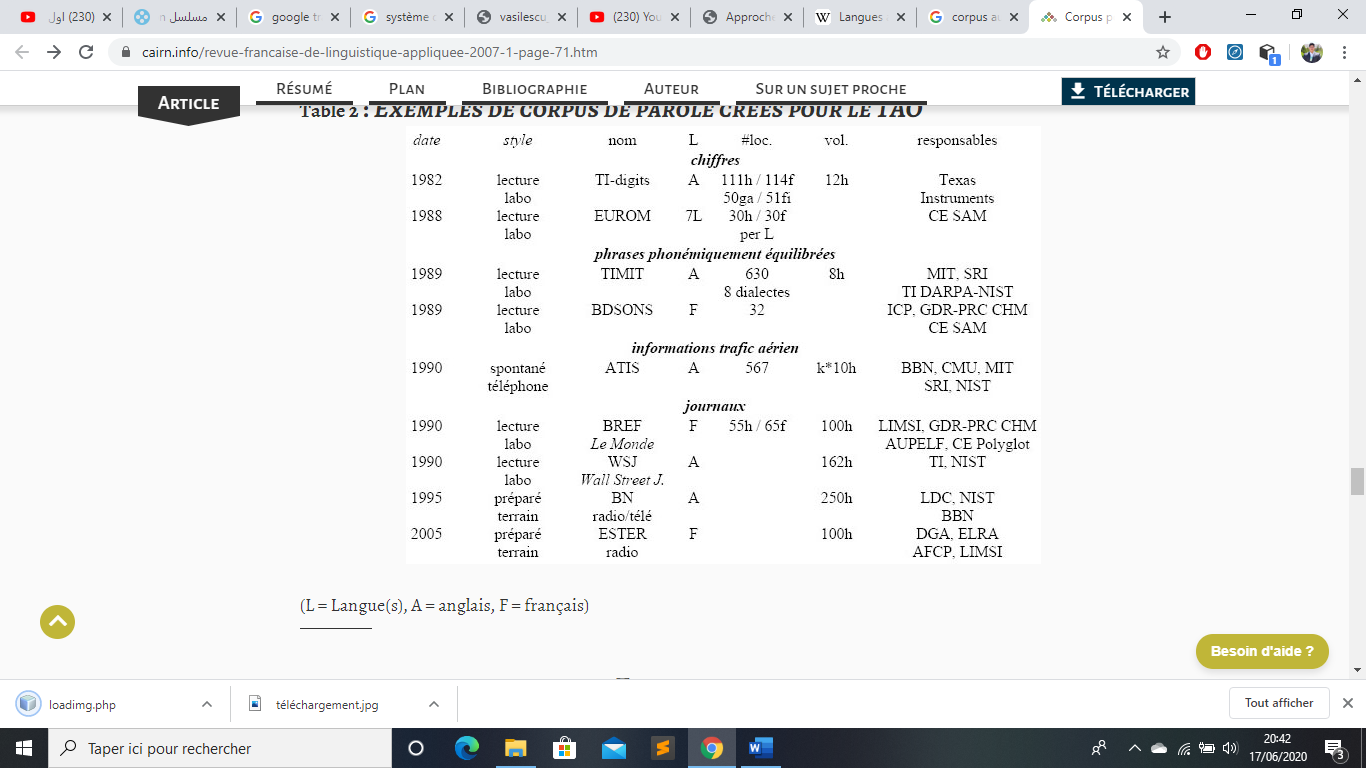
ATIS [20] Les données de test ATIS3 contiennent le matériel de développement, de test et d'évaluation utilisé dans les tests de référence de décembre 1994 dans le cadre du projet ATIS (Air Travel Information Services). Cette version contient également la base de données relationnelle ATIS3 46 villes / 52 aéroports.

BREF [21] Le corpus BREF a été conçu pour fournir suffisamment de données vocales lues pour le développement et l'évaluation de systèmes de reconnaissance vocale continue (à la fois dépendant du locuteur et indépendant du locuteur), et pour fournir un grand corpus de parole continue pour l'acquisition de connaissances acousto-phonétiques de parlé français.L'ensemble du corpus BREF contient plus de 100 heures de discours de 120 locuteurs.

WSJ [22] contient une version complète, de style Treebank, de la partie du discours (POS) étiquetée et analysée de la collection de trois ans du Wall Street Journal (WSJ) d'ACL / DCI ( [LDC93T1](https://catalog.ldc.upenn.edu/LDC93T1) ), environ 30 millions de mots.

BN [23] Le British National Corpus (BNC) a été créé à l'origine par la presse de l'Université d'Oxford dans les années 1980 - début des années 1990, et il contient 100 millions de mots de textes de textes provenant d'un large éventail de genres (par exemple, parlé, fiction, magazines, journaux et universitaires).

ESTER [24] Le corpus ESTER est un sous-ensemble du package d'évaluation ESTER (réf. Catalogue ELRA-E0021), produit dans le cadre du projet national français ESTER dans le cadre du programme Techno langue financé par le ministère français de la Recherche et des Nouvelles Technologies (MRNT). Le projet ESTER a permis de mener une campagne d'évaluation des systèmes de transcription enrichis de Broadcast News à partir de données françaises. Ce corpus comprend le matériel qui a été utilisé pour la campagne d'évaluation ESTER, à l'exclusion des données textuelles Environ 100 heures de diffusion de nouvelles transcrites orthographiquement, y compris des annotations d'entités nommées.



(L = Langue(s), A = anglais, F = français)

Figure 6 : Exemples de corpus de parole Européens et Anglais

1. Exemple de corpus audios qui traitent l’identification de parole des dialectes arabes [5]:

Dans cette section, nous allons citer des corpus audios qui traitent les dialectes arabes. Bien qu’il existe peu de corpus pour l’arabe, c’est dernier est l’une des langues les plus répandues dans le monde.

Ces corpus sont classés en cinq catégories selon la méthode de collecte, cela peut se faire en enregistrant des conversations téléphoniques spontanées, des réponses téléphoniques de questionnaires, des enregistrements directs et des ressources sur le Web.

La figure 7 [6] présente certaines caractéristiques des corpus d’arabes dialectaux étudiés.

Le premier corpus, c’était de l’Egypte qui s’appelle CALLFRIEND [7] qui comprend 60 conversations téléphoniques non programmées, d'une durée comprise entre 5 et 30 minutes. Le corpus comprend également une documentation décrivant l'information du locuteur (sexe, âge, éducation, numéro de téléphone de l'appelé) et des informations d'appel (qualité du canal, nombre de locuteurs). Une autre partie du projet OrienTel [8], citée ci-dessous, a été consacrée à la collecte de corpus de discours pour les dialectes arabes d'Egypte, de Jordanie, du Maroc, de Tunisie et des pays des Emirats Arabes Unis.

Le corpus DARPA Babylon Levantine [9] Arabe regroupe quatre dialectes Levantins parlés par des locuteurs de Jordanie, de Syrie, du Liban et de Palestine (Makhoul et al. 2005).

La Appen’s corpora [10] a rassemblé trois corpus de dialectes arabes au moyen de la méthode des conversations téléphoniques spontanées. Ces corpus 3 ont été prononcés par des orateurs du Golfe, de l'Iraq et du Levant. Avec un protocole d'enregistrement de conversation téléphonique plus guidé, le corpus arabe de Levant est disponible via le catalogue 4 des PMA. Les conférenciers sont choisis en Jordanie, au Liban, en Palestine, au Liban, en Syrie et dans d'autres pays du Levant.

TuDiCoI [11] est un corpus de dialogue spontané dédié au dialecte tunisien, qui contient des dialogues enregistrés entre le personnel et les clients du chemin de fer de la ville de Sfax, en Tunisie.

SAAVB [12] corpus est dédié aux locuteurs de toutes les villes du pays Arabie saoudite en utilisant la réponse par téléphone de la méthode du questionnaire. La caractéristique principale de ce corpus est que, avant l'enregistrement, un choix préliminaire de locuteurs et d'environnement est effectué. La sélection vise à contrôler l'âge, le sexe et le type de téléphone du locuteur.

KSU [13] Le corpus arabe riche comprend des locuteurs de différents groupes ethniques, arabes et non-arabes (Afrique et Asie). Concernant les locuteurs arabes de ce corpus, ils sont sélectionnés parmi neuf pays arabes : Arabie saoudite, Yémen, Egypte, Syrie, Tunisie, Algérie, Soudan, Liban et Palestine. Ce corpus est riche en plusieurs aspects. Parmi eux, la richesse du texte d'enregistrement. En outre, différentes sessions d'enregistrement, environnements et systèmes sont pris en compte.

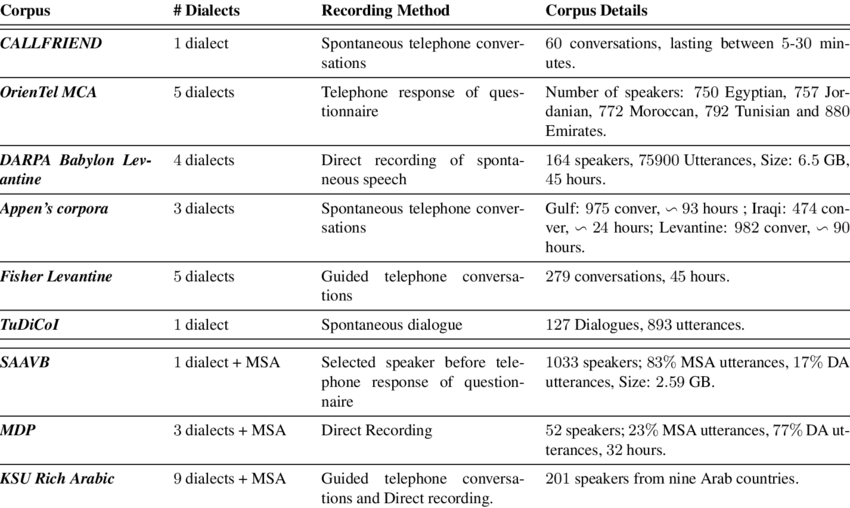


Figure 1 : Caractéristiques des corpus d’arabes dialectaux étudiés

1. Exemple de corpus audios qui traitent les dialectes arabes (ADI17) [15]:

Arabic Dialect Identification (ADI17) contient plus d'un million de segments de discours pour 17 pays arabes, extraits de vidéos téléchargées sur YouTube.  
La tâche de l'ADI est l'identification dialectale de la parole des 17 dialectes d’après les vidéos téléchargées sur YouTube. Les études précédentes sur l'identification des dialectes arabes sont limitées à 5 classes de dialectes.

* L'arabe égyptien (EGY) couvre les dialectes de la vallée du Nil: l'Égypte et le Soudan.
* L'arabe levantin (LAV) comprend les dialectes du Liban, de la Syrie, de la Jordanie et de la Palestine.
* L'arabe du Golfe (GLF) comprend les dialectes du Koweït, des Émirats arabes unis, de Bahreïn et du Qatar. L'Arabie saoudite est généralement incluse, bien qu'il y ait un large éventail de sous-dialectes en son sein. L'arabe omanais est parfois également inclus.
* L'arabe de l'Afrique du Nord (NOR) - également connu sous le nom de Maghrebi - couvre les dialectes du Maroc, de l'Algérie, de la Tunisie et de la Mauritanie. L'arabe libyen est parfois également inclus.
* L'arabe standard moderne (MSA), qui constitue un discours formel.

La classification ADI (Arabic Dialect Identification) suppose que chaque segment de parole correspond à un dialecte.

Il existe trois éditions des données de défi ADI Challenge :

* InterSpeech 2016 [forme](https://github.com/qcri/dialectID/tree/master/data/train.IS2016) et [teste les](https://github.com/qcri/dialectID/tree/master/data/test.IS2016) données.
* MGB-3 / Vardial 2017 [train](https://github.com/qcri/dialectID/tree/master/data/train.vardial2017) , [dev](https://github.com/qcri/dialectID/tree/master/data/dev.vardial2017) et [test](https://github.com/qcri/dialectID/tree/master/data/test.MGB3) data.
* [Vardial 2018.](http://alt.qcri.org/vardial2018/)

Pour présenter une analyse fine du discours du dialecte arabe, la collecte des données est faite sur YouTube.

Pour **Train** set, environ 3 000 heures de données ont été collectées sur YouTube. Parfois certains ensembles de données pourrait contenir des erreurs d’étiquetage, pour cette raison, nous avons deux sous-pistes pour la tâche ADI, une piste d'apprentissage supervisé et une piste non supervisée.

Pour l'ensemble **Dev** et **Test**, environ 280 heures de données vocales ont été collectées sur YouTube. Après la liaison automatique des locuteurs et l'étiquetage du dialecte par des annotateurs, 57 heures de données vocales ont été sélectionné pour l’utiliser comme ensemble de développement et de test pour l'évaluation des performances. L'ensemble de données de test a été considéré comme ayant trois sous-catégories selon la durée du segment, pour représenter la durée courte (moins de 5 secondes), moyenne (entre 5 secondes et 20 secondes), longue (plus de 20 secondes) du discours dialectal.

* 1. Exemple d’un site d’identification de langue arabe

L'identification du dialecte arabe (ADI) est similaire au problème plus général de l'identification des langues (LID). L'ADI est plus difficile que le LID en raison des petites et subtiles différences entre les divers dialectes d'une même langue. Un bon système ADI peut être utilisé pour extraire des données dialectales de la base de données de la parole afin de former des modèles acoustiques spécifiques au dialecte pour la transcription de la parole en texte. Il peut également être utilisé pour l'enrichissement des métadonnées.  
Ce système d'identification de dialecte arabe en direct ; QCRI-MIT Système avancé d'identification des dialectes (QMDIS). Ce projet propose des technologies Web modernes pour capturer l'audio en direct et diffuse simultanément des transcriptions en arabe avec le dialecte correspondant. Le site de la démo est accessible au public sur ***dialectid.qcri.org***.

1. Exemple avant la capture de l’audio :

La tâche de l'identification des dialectes parlés consiste à classer un énoncé parlé dans l'un des nombreux dialectes d'une langue particulière. La langue arabe peut être largement divisée en cinq dialectes principaux ; à savoir l'Égypte (EGY), le Golfe (GLF) ou la péninsule arabique, le Levantin (LAV), l'arabe standard moderne (MSA) et l'Afrique du Nord (NOR) ou les pays du Maghreb.

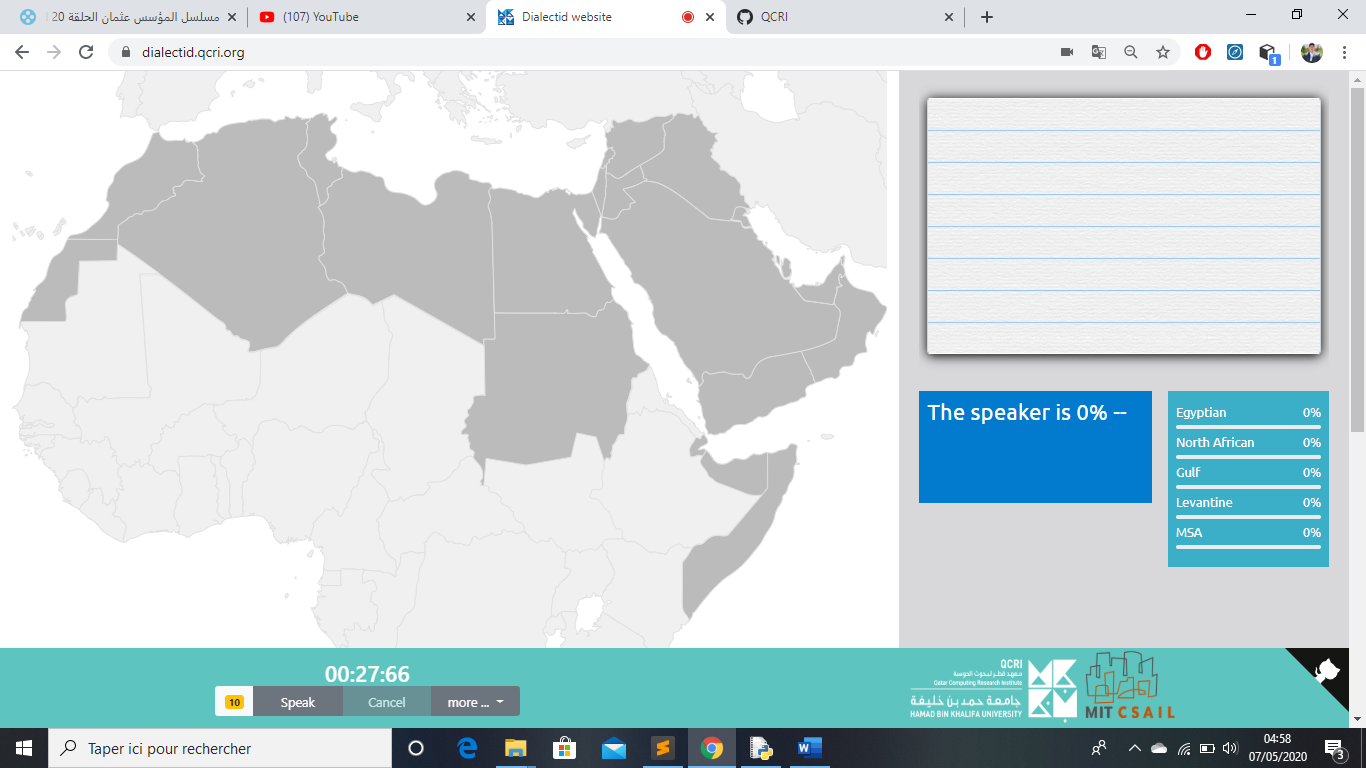


Figure 8 : Avant la capture de l’audio

1. Exemple après la capture de l’audio et la détection du dialecte :

Cette interface capture l'audio en direct et diffuse simultanément des transcriptions en arabe avec le dialecte correspondant. Le dialecte détecté est visualisé à l'aide d'une carte lumineuse, où l'intensité de la couleur reflète la probabilité du dialecte. Elle affiche également des barres de mesure pour la probabilité de chaque dialecte par phrase

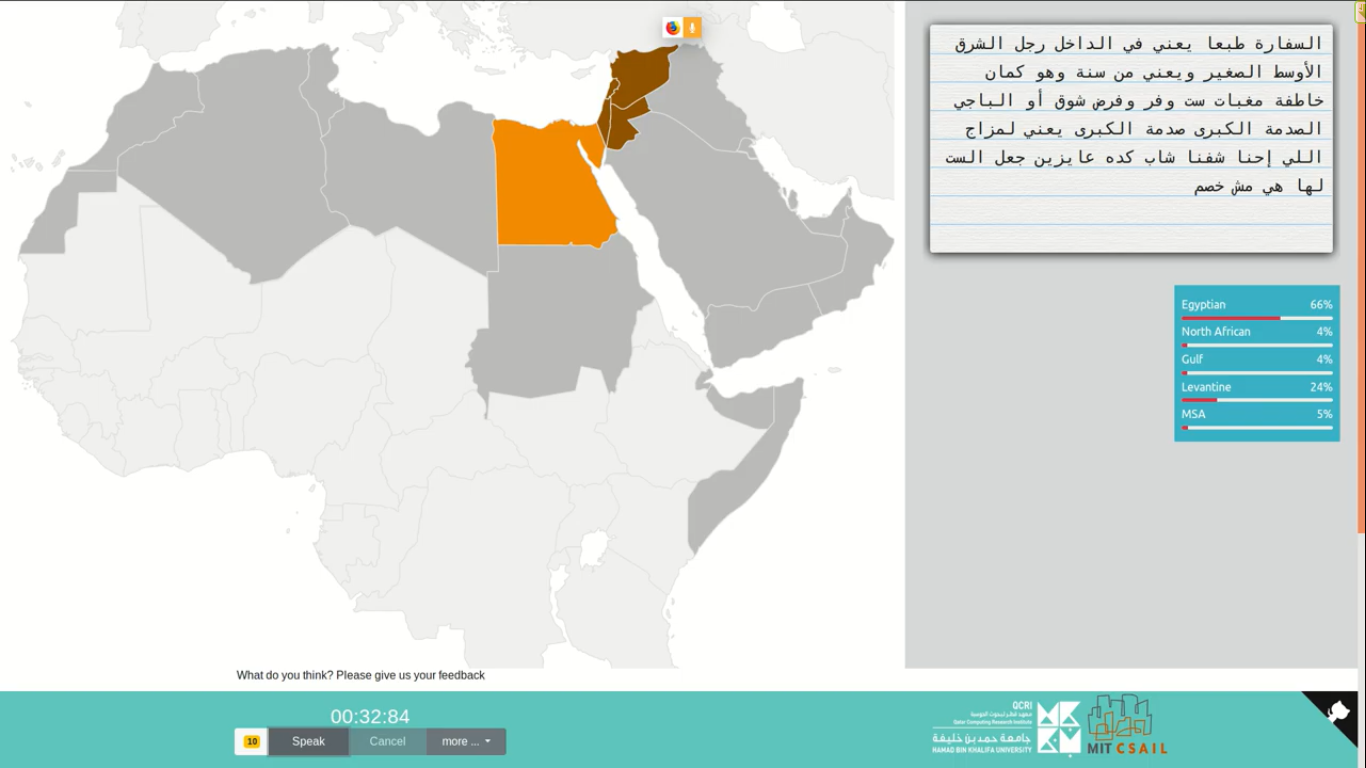


Figure 9 : Après la capture de l’audio

* 1. Outils de travail

Les outils utilisés pour la réalisation de ce projet sont :

* + 1. YouTube API (Pour la recherche des vidéos sur YouTube)



Figure 10 : Logo YouTube Data API

L'API YouTube permet aux développeurs d'accéder aux statistiques vidéo et aux données des chaînes YouTube via deux types d'appels, REST et XML-RPC. Google décrit les ressources de l'API YouTube comme des « API et outils qui vous permettent d'intégrer l'expérience YouTube à votre page Web, votre application ou votre appareil ». Ceci est l'un des développeurs Google.

2.5.2 InaSpeechSegmenter :

InaSpeechSegmenter est une boîte à outils de segmentation audio basée sur CNN. Il divise les signaux audios en zones homogènes de parole, de musique et de bruit. Les zones vocales sont divisées en segments étiquetés en fonction du sexe du locuteur (masculin ou féminin). Les modèles de classification des hommes et des femmes sont optimisés pour la langue française car ils ont été formés en utilisant des francophones (les corrélats acoustiques du sexe des locuteurs dépendent de la langue). Les zones correspondant à la parole sur la musique ou la parole sur le bruit sont étiquetées comme parole.

InaSpeechSegmenter a été conçu pour réaliser [des études](http://doi.org/10.18146/2213-0969.2018.jethc156) à [grande échelle sur l'égalité des sexes](http://doi.org/10.18146/2213-0969.2018.jethc156) basées sur l'estimation du pourcentage de temps de parole des hommes et des femmes.

2.5.3 Pydub :



Figure 12 : Logo PyDub

Sert à manipuler les audio avec une interface de haut niveau. Il est simple et facile.

2.5.4 FFmpeg :

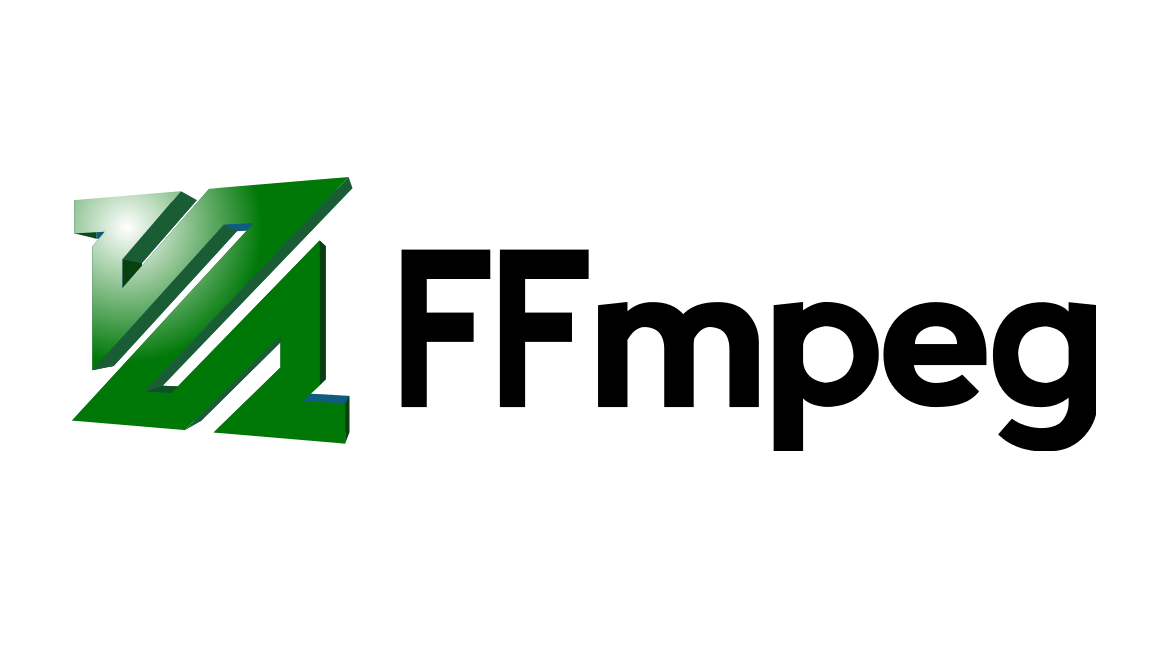


Figure 13 : Logo FFmpeg

FFmpeg est une collection de [logiciels libres](https://fr.wikipedia.org/wiki/Logiciel_libre) destinés au traitement de flux audio ou vidéo (enregistrement, lecture ou conversion d'un format à un autre).Ce projet a été créé par [Fabrice Bellard](https://fr.wikipedia.org/wiki/Fabrice_Bellard) en [2000](https://fr.wikipedia.org/wiki/2000), et est maintenant maintenu par Michael Niedermayer. Il est actuellement hébergé sur les serveurs du projet [MPlayer](https://fr.wikipedia.org/wiki/MPlayer). Il a été développé sous [GNU/Linux](https://fr.wikipedia.org/wiki/GNU/Linux) mais peut cependant être compilé sur la plupart des [systèmes d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation), y compris [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows). Le choix de la licence [GNU LGPL](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_limit%C3%A9e_GNU) ou GPL dépend du choix de configuration d'options.

2.5.5 Sox :

1. La collecte d’un corpus audio depuis le web pour les dialectes marocains
   1. Introduction

La tâche d'identification des dialectes est un cas particulier du problème plus général de l'identification des langues. Elle fait référence au processus d'identification automatique de la classe de langue pour un segment vocal ou un document texte donné. Pour notre projet on va travailler juste sur les données audios. La langue Marocaine a plusieurs dialectes parlés. Il existe sept principaux dialectes, pour l'arabe dialectal on a l’Urbain, l’Arabe des montagnes du Nord-Ouest, le Bédouin de l’Ouest ainsi que celui de l’Est et la Hassanya. Et pour

l’Amazigh, on a le Tachelhit et le Tarifit.

L'identification des dialectes marocains est sans doute un problème difficile, car elle consiste à identifier les différents dialectes au sein d'une même classe de langue. Ainsi, l'identification automatique du dialecte d'entrée à partir du signal vocal a été un problème de recherche intéressant à la fois en soi et pour améliorer la reconnaissance automatique de la parole.

Pour cela, nous avons adopté une stratégie qui nous permettra de nous comparer aux corpus existants, et de disposer d'un corpus suffisamment solide pour la formation d'un modèle d'apprentissage machine et aussi pour un grand système d'identification des dialectes marocains, vous trouverez ci-dessous un schéma de récapitulation :

Figure 2 : Etapes de collectes

* 1. Les étapes de la construction du corpus audio
     1. L’inventaire des sources vidéo

Plus qu’on a un grand nombre de vidéos, plus il y a beaucoup de types de vidéos tels que le journal quotidien, météo, cuisines, sport,… Or, notre but n’est pas de citer les types de vidéos basiques qu’on connait très bien, l’idée est d’identifier justement des types qui vont nous aider à différencier une vidéo d’une autre selon leurs métadonnées en dépendance de la culture marocaine ce qui fait qu’on va avoir un robuste DataSet.

Comme exemple :

**- Nouvelles locales** : il y a des chaines qui fournissent les nouvelles d’une certaine ville et ses environs par exemple « ChoufTv ,KifachTv» .

**- Films marocains ou amazighs**

**- Séries marocaines ou amazighs**

**- Des interviews**

**- Des émissions télévisées**

**- La radio**

**- …**

* + 1. La recherche des chaines IDs :

Comme première étape, il faut bien choisir les chaines sur YouTube car **La collecte des chaines se fait manuellement donc le bon choix des chaines nous permet de générer un robuste dataset ,pourtant il est simple de caractériser les chaines IDs sur YouTube, d’après l’URL .Par la suite on doit remplir le fichier csv avec l’ID de la chaine et le dialecte cible sépares par une virgule ,vous trouverez ci-dessous des exemples des chaines cibles.**

**Dialectes Nom des chaines Abréviation**

Urbains **Kifach.tv** URB

**MFM Radio**

**TopTivi**

**Chada FM**

Montagnes **LinamSolution** MNO

du Nord-Ouest

**HM Tv**

Bédouins de l’Ouest **MFM Radio Maroc** BEO

**2M-Hassan EL Fad**

**Baraka El Baraka**

**Chafik production**

Bédouins de l’Est **Kamal Taibi**  BEE

**Abdelkader secteur**

Hassanya **chaine HAYAT essahrawiya** HAS

Tarifit **Yan Production**  TAR **قناة الريفية - rifiya TV**

Tachelhit **chaine officiel Aflam Hilal Vision** TAC

Figure 14 : Exemple chaines IDs

* + 1. Filtrage des chaines

Après avoir obtenu une liste complète des chaines qui répond à notre besoin concernant les différents dialectes c.-à-d. il contient des vidéos parlant un parmi les 7 dialectes, On a essayé de faire un sort de filtrage manuelle qui se base sur le nombre des playlists parlant le dialecte cible, la qualité de son et aussi le nombre de vidéos intrus c.-à-d. qui parlent un autre dialecte, $$$$$$$$$$

* + 1. Mise en place de l'environnement

YouTube étant une filiale de Google, les différentes API YouTube sont fournies et maintenue par Google. Pour accéder à une API fournie par Google, les étapes suivantes sont nécessaires :

1. Vous aurez besoin d'un compte Google pour accéder à la console développement Google.
2. Créez un nouveau projet dans votre console de développement Google.
3. Vous pouvez ensuite demander une clé API et enregistrer votre application.
4. Les API requises auxquelles vous devrez accéder devront être activées dans la bibliothèque de l’API de Google Cloud Console [25].

Une fois la clé API obtenue et configurée, vous pouvez l'utiliser pour passer des appels à l'api.

Pour les appels API généraux, tout ce dont vous avez besoin est la clé API.  
Mais si vous avez besoin d'accéder à un compte d'utilisateur pour obtenir plus d'informations, télécharger ou modifier les données d'un compte d'utilisateur, vous aurez besoin d'un « identifiant client » et un « secret client » Ceux-ci sont utilisés pour l'authentification avec le Framework oauth2 [26].

Puisque nous allons travailler dans une environnement Python nous avons besoin de :

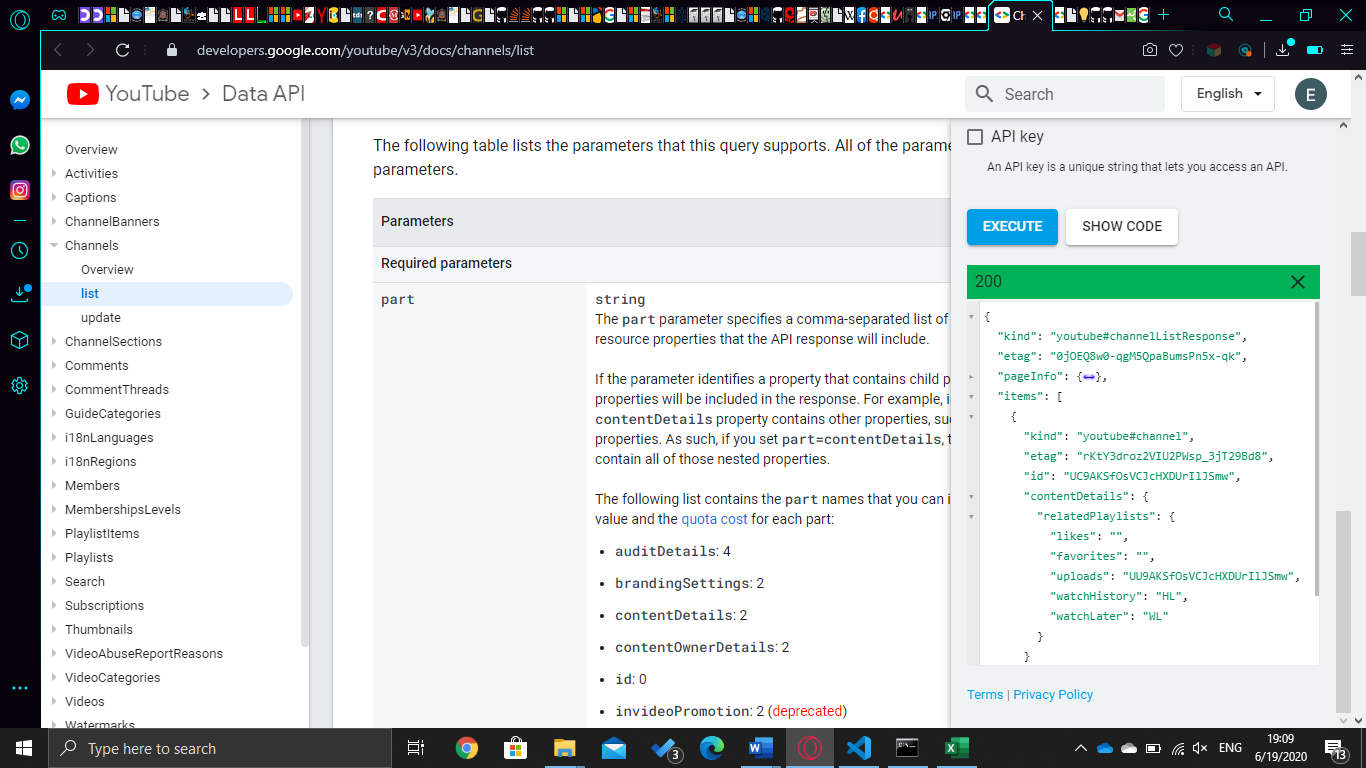
* Python 2.7 ou Python 3.5+.
* L'outil de gestion des paquets pip.
* L'API Google pour la bibliothèque client Python et les bibliothèques « google-auth-oauthlib » et « google-auth-httplib2 » pour l'autorisation des utilisateurs. [27]
  + 1. L’extraction des IDs vidéos

Afin d’avoir un nombre suffisant de vidéos on est amené à extraire tous les vidéos de chaque chaine dans la liste obtenue au préalable pour cela on a écrit un script permettant d’exploiter à la fois les fonctionnalités de YouTube data api et le langage python.

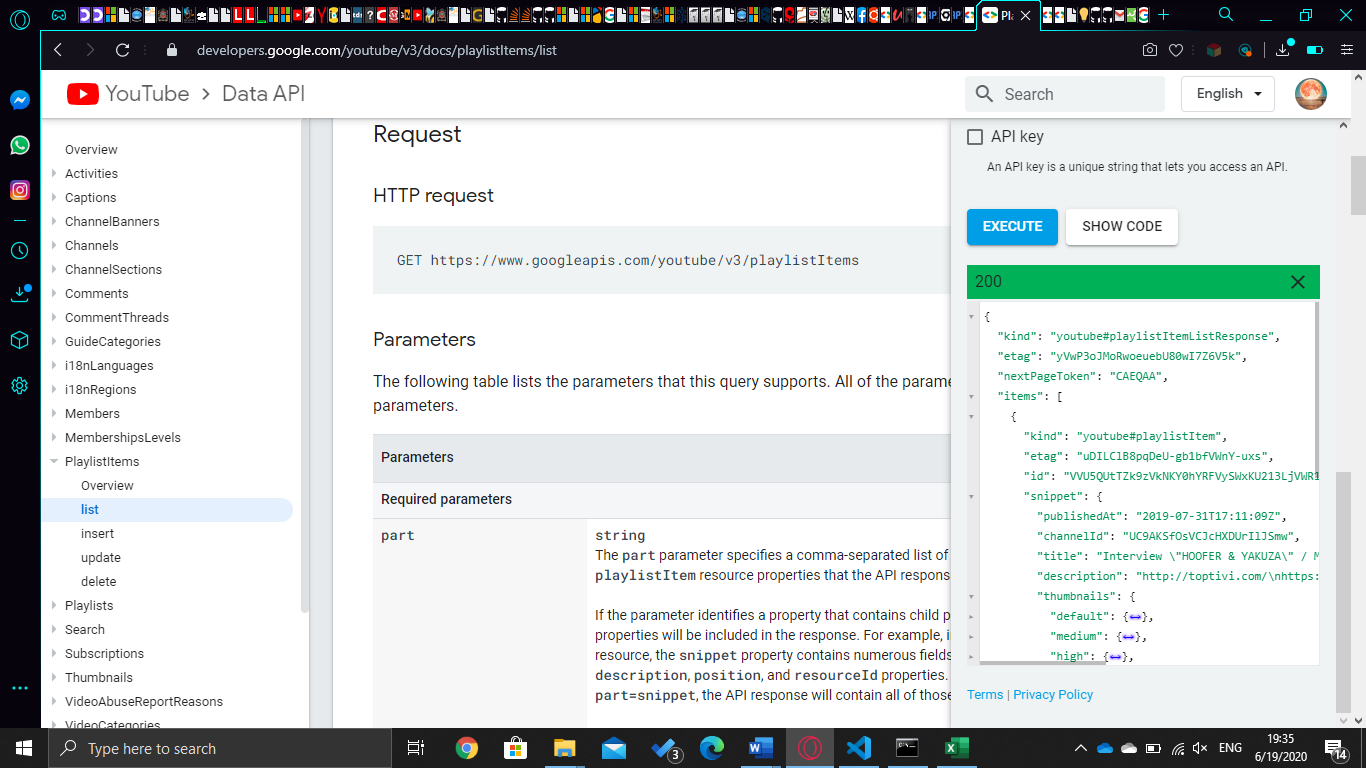
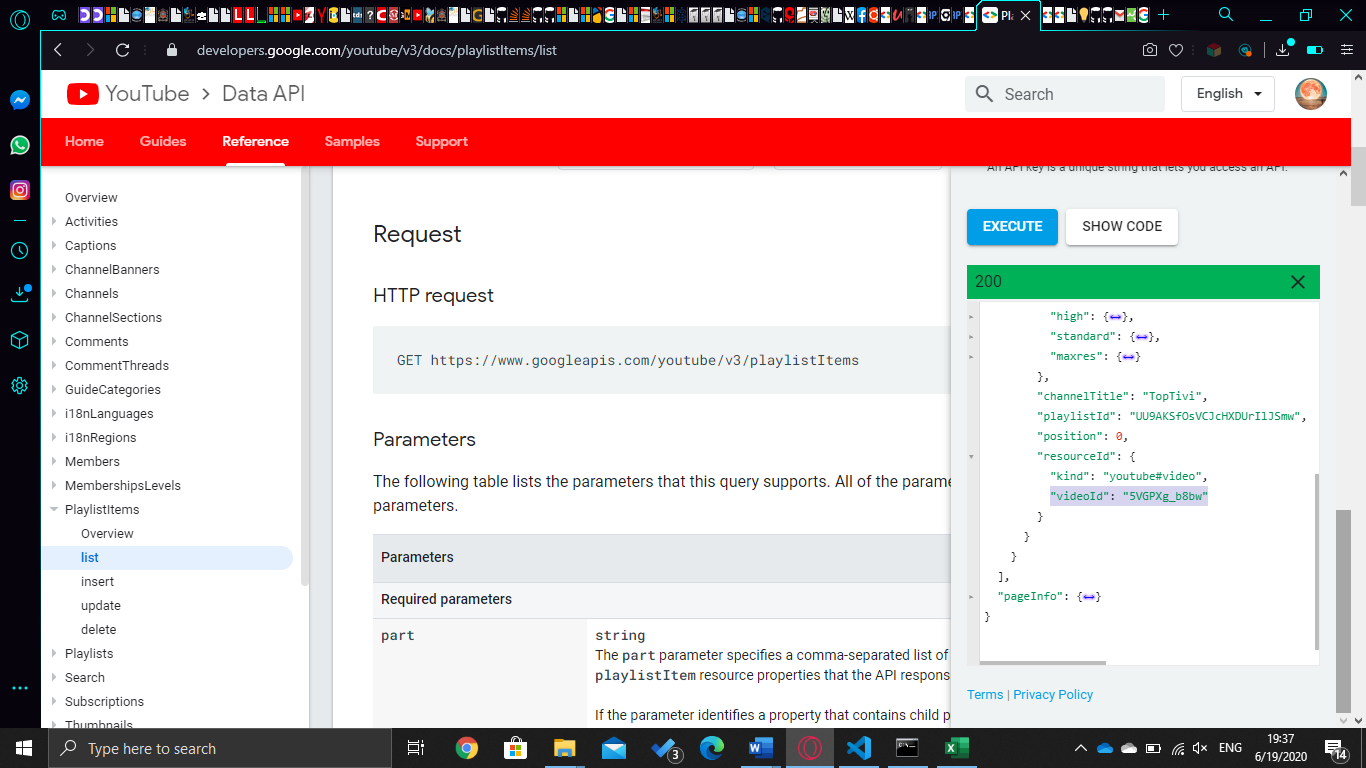
Le fonctionnement principal de ce script est comme suit :

1. La lecture des identifiants des chaines sélectionnées a partir du fichier csv générer au paravent.
2. Pour chaque identifiant de chaîne on envoie une première requête http vers l’Api qui va nous retourner la description de la chaine sous format d’un dictionnaire python, si le résultat est vide cela signifie que l’identifiant courant n’est pas vraiment un identifiant YouTube de cette chaine c’est juste un nom d’utilisateur qui a été choisi par le propriétaire de la chaîne. Pour ce dernier cas il faut donc envoyer une autre requête qui va nous retourner le vrai identifiant de la chaîne.

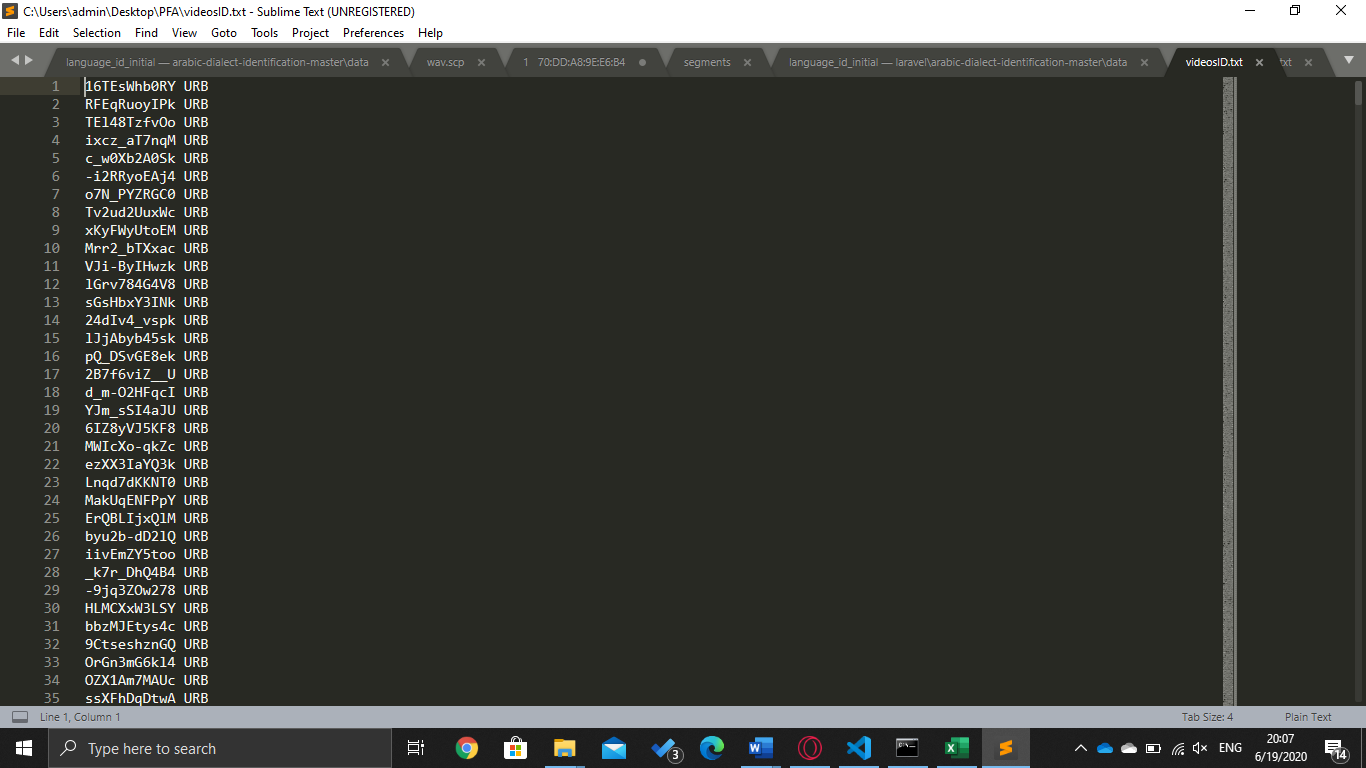
Voici un exemple de résultat de la requête :



1. Après que le résultat du première requête soit prête on va l’exploiter pour envoyer une deuxième requête qui vas nous donner la liste de tous les vidéos de la chaine courante, cela est fait comme suit :
   * + - * La propriété « uploads » dans la description de la chaine représente l’identifiant de la liste de lecture de tous les vidéos de la chaine.
         * En exploitant la propriété « uploads » on envoie une autre requête vers l’Api qui va nous retourner la liste totale des identifiant de tous les vidéos de la chaine courante.

Voici un exemple du résultat obtenu :

1. Après que la liste des identifiants de tout le vidéo est prête il nous reste que les écrire dans un fichier texte avec le label du dialecte correspondant à chaque vidéo. Le label de chaque chaine et le même pour tous ces vidéos.

Voici un exemple d’après le fichier obtenu qui contient exactement 15851 identifiants :

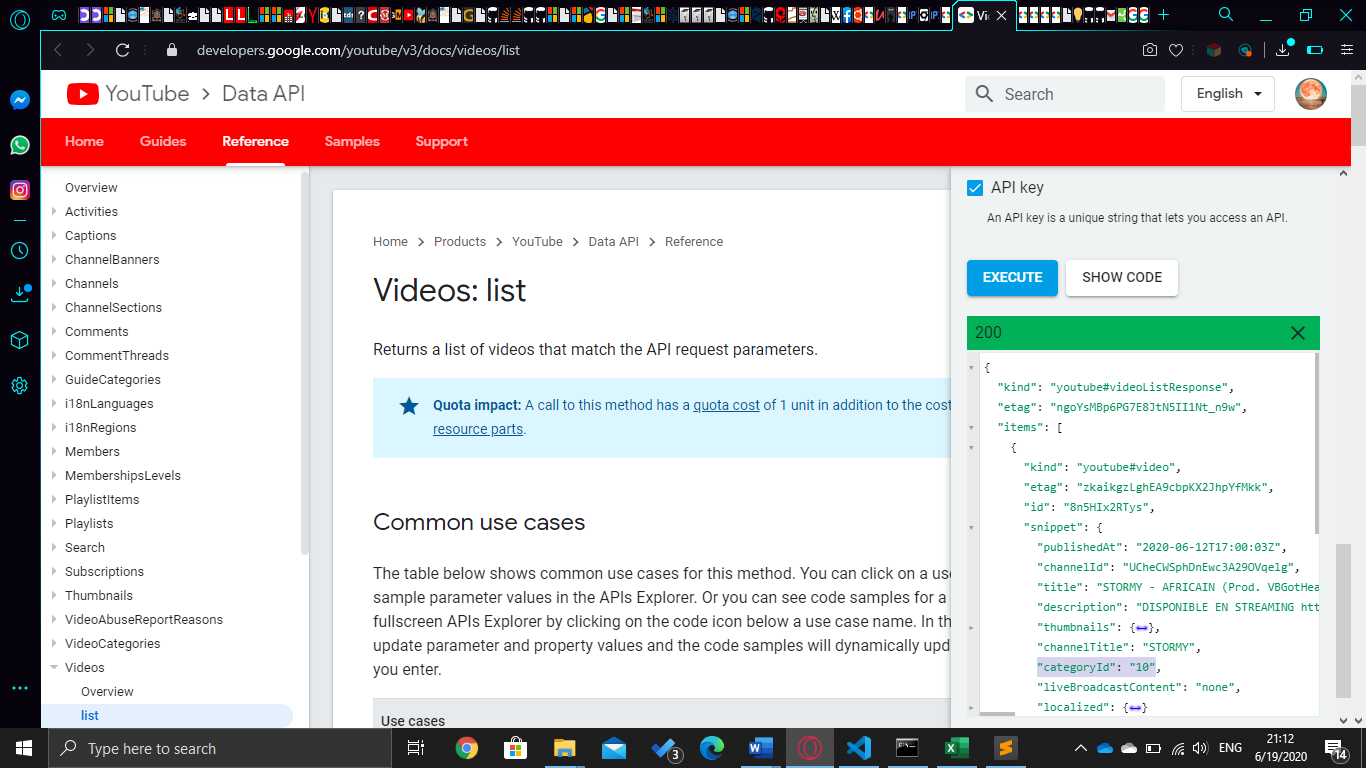
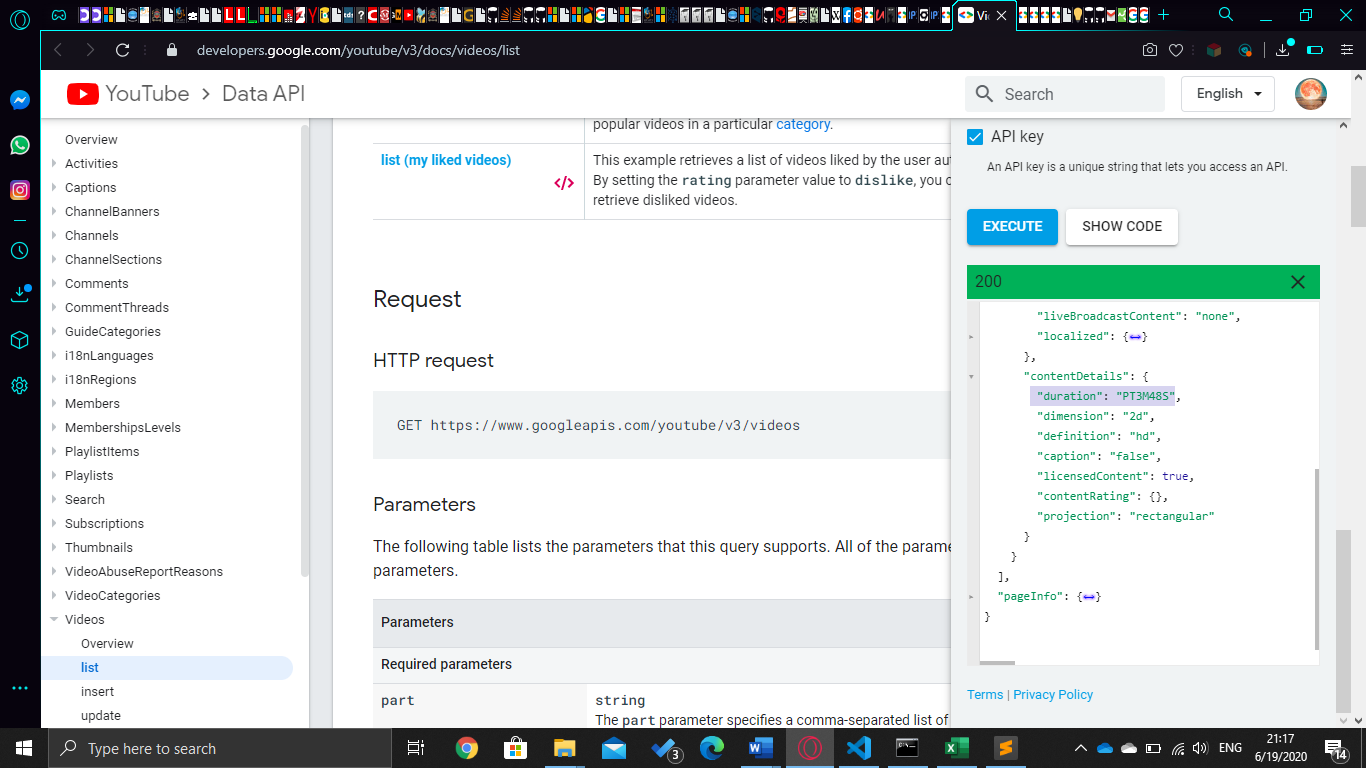
* + 1. L’extraction des informations des audios

Afin d'avoir un maximum d'informations sur toutes les vidéos obtenues, nous avons pensé à écrire un second script python basé sur le même principe que le premier et qui exploite également YouTube Data Api.

Le script fonctionne comme suit :

1. La liste des identifiants des vidéos et un dictionnaire python contenant la liste des catégories et leurs identifiants ces catégories [28] sont publier par YouTube, sont les deux entrées principales du programme.
2. L’envoi d’une requête http vers l’api qui va nous retourner la description de chaque vidéo de la liste obtenu au préalable.

Voici un exemple du résultat :



1. D’après la description on est intéressé juste par la durée et l’identifiant de catégorie, pour cela le script prend juste ces deux informations.
2. Pour définir la vraie catégorie de chaque vidéo on a exploité le dictionnaire présenter dans la première étape, et pour la durée on appeler une simple fonction qui fait le convertissement en secondes.
3. Par la suite le scripte se charge aussi d’écrire les résultats dans un nouveau fichier ou les ligne sont sous la forme (identifiant, label du dialecte, catégorie, durée)

Voici un extrait du fichier obtenu :

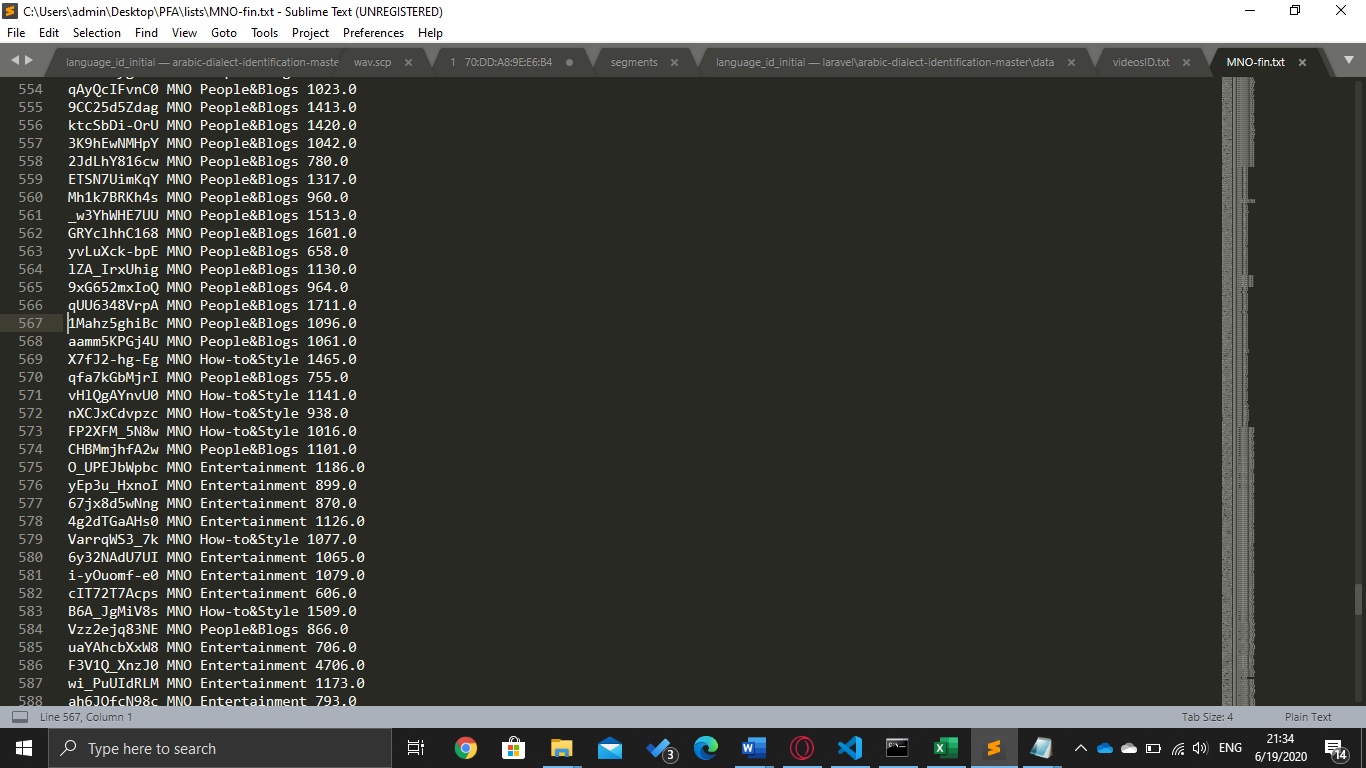


Figure 15 : Vidéos IDs, dialecte label, genre et durée

* + 1. Filtrage des audios

Avant de passer à l’extraction des audios et sur le concept de bien améliorer l’ensemble de données on a fait un filtrage des audios manuellement, pour chaque 20 audios on choisit une et on la teste sur YouTube, si elle va nous servir on garde tous les 20 vidéos sinon on les supprime, cela est dû à le fait que les 20 vidéos sont fort probable du même genre et appartiennent à la même playlist, et aussi on a décidé de supprimer tous les vidéos du genre musique à l’aide d’un script python qui prend en entrée la liste obtenu au préalable et il vérifie si l’identifiant correspond à un vidéo de genre musique ou non ,si oui il supprime l’identifiant courant.

* + 1. L’extraction des audios

Afin d’automatiser la tache de téléchargement on a introduit un autre script qui se base sur le Framework FFMPEG et YouTube-dl qui est un programme en ligne de commande permettant de télécharger des vidéos à partir de YouTube et de quelques autres sites. Il nécessite l'interpréteur Python, version 2.6, 2.7 ou 3.2+, et il n'est pas spécifique à la plate-forme. Cela devrait fonctionner sur votre box Unix, sur Windows ou sur MacOs. Il est rendu public, ce qui signifie que vous pouvez le modifier, le redistribuer ou l'utiliser comme bon vous semble.

Le script se base aussi sur SOX qui est un utilitaire de ligne de commande qui peut convertir différents formats d'ordinateur fichiers audio dans d'autres formats. Il peut également appliquer divers effets à ces fichiers audios lors de la conversion. En prime, SOX peut jouer et enregistrer les fichiers audios sur plusieurs plates-formes, il est aussi capable de gérer des formats comme MP3, WAV, AIFF, etc.

Après le filtrage de la liste des vidéos on a une large base de données prête à collecter, notre script python, qui prend en entrée la liste des identifiant vidéos obtenu au préalable, et basée sur le téléchargement de chaque vidéo de la liste en format MP4 en utilisant « YouTube-dl », après que ce fichier est présent dans le répertoire le programme se charge de le transformer en format WAV en exploitant les deux bibliothèque « FFMPEG » pour la convertissement MP4 en format WAV et « SOX  » pour fixer le nombre d’échantillons par seconde en 16000 afin d’améliorer au maximum la qualité des audios et avoir une base de donnée homogène en terme de nombre d’échantillon par seconde ou ce qu’on appelle le fréquence d’échantillonnage [30].

**Pourquoi utiliser le format WAV ?**

Le format WAV est très populaire dans le **monde de la musique** grâce à la qualité d’écoute qu’il propose. Il ne compresse pas les fichiers audios et leur permet donc de garder une excellente qualité. Résultat, c’est le format qui est utilisé par presque tous les logiciels de création et de mixage audio professionnels.

Si vous souhaitez travailler sur un logiciel de ce genre, pour une raison ou pour une autre, vous allez donc souvent d’abord devoir convertir vos fichiers audio (MP3, WMA, etc.). Vous trouverez plus d’informations à ce sujet [**sur cette page de l’éditeur Movavi**](https://www.movavi.com/fr/support/how-to/comment-convertir-fichier-audio-en-wav.html) [29]. Cela vous permettra de convertir n’importe quel fichier au format WAV et ainsi de l’utiliser dans des conditions professionnelles pour réaliser ce que vous souhaitez.

A cause des droits d'auteur la majorité des vidéos du dialecte TAC (Tachelhit) ont été bloquées lors du téléchargement et on a réussi à télécharger juste environ 20 heures pour ce dialecte.

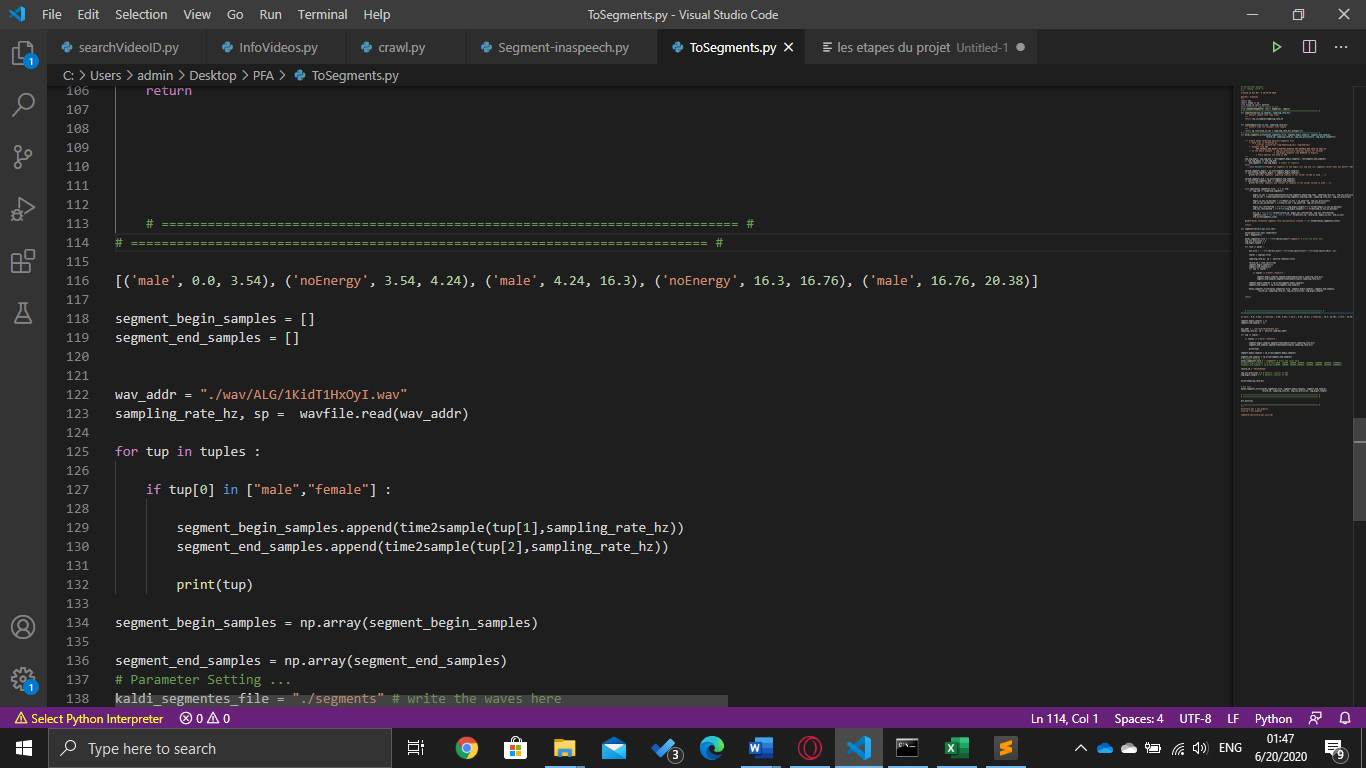
* + 1. La segmentation des audios

Cette partie nous permet de faire la segmentation théorique. C’est le fait de découper les audios en segments avec le Framework en python InaSpeechSegmenter, d’abord on doit expliquer son principe, InaSpeechSegmenter est une boîte à outils de segmentation audio basée sur CNN[31]. Il divise les signaux audios en zones homogènes de parole, de musique et de bruit. Les zones vocales sont divisées en segments étiquetés en fonction de temps de début et temps de fin de parole, musique ou silence.

 La classe de INASPEECHSEGMENTER permettant d'effectuer des segmentations s'appelle Segmenter. C'est la seule classe qui doit être importer dans un programme. Le constructeur de classe accepte 3 arguments facultatifs :

* Vad\_engine (par défaut : 'smn'). Permet de choisir entre 2 moteurs de détection d'activité vocale.
* 'smn' est le moteur le plus récent et divise le signal en segments de parole, musique et bruit
* 'sm' n'a pas été formé avec des exemples de bruit et a divisé le signal en segments de parole et de musique. Les segments de bruit sont considérés comme de la parole ou de la musique. Ce moteur a été utilisé dans l'étude ICASSP et a remporté le défi de détection de la parole MIREX 2018.
* Detect\_gender (par défaut : True): s'il est défini sur True, effectue la segmentation par sexe sur le segment de la parole et génère les étiquettes «femelle» ou «mâle». Sinon, sort les étiquettes « discours » (plus rapidement).
* Ffmpeg : permet de fournir un binaire spécifique de ffmpeg au lieu de l'installation système par défaut.

Le script se charge d’extraire les segments de parole de chaque audio en passant par ces étapes :

1. Les fichiers « WAV » sont organiser de cette manière : Pour chaque dialecte on trouve un dossier qui porte le même nom du dialecte et qui contient tous les fichiers correspondant à ce dialecte.
2. L’organisation des fichiers « WAV » dans notre répertoire nous facilite l’accès vers tous les fichiers par le moyen de ce script. Donc pour deviner le chemin on a besoin juste du fichier des identifiant et labels.
3. Le Framework « INASPEECH SEGMENTER » nous permet de générerUne liste des tuples pour chaque fichier « WAV », voici un exemple :
4. Chaque tuples (Male, Female) va représenter notre segment de parole.

Par la suite le programme charge d’écrire le nom de chaque segment dans un nouveau fichier sous la forme (identifiant\_début-en-ms\_fin-en-ms identifiant-audio début-en-sec fin-en-sec)

Voici un extrait du fichier obtenu :

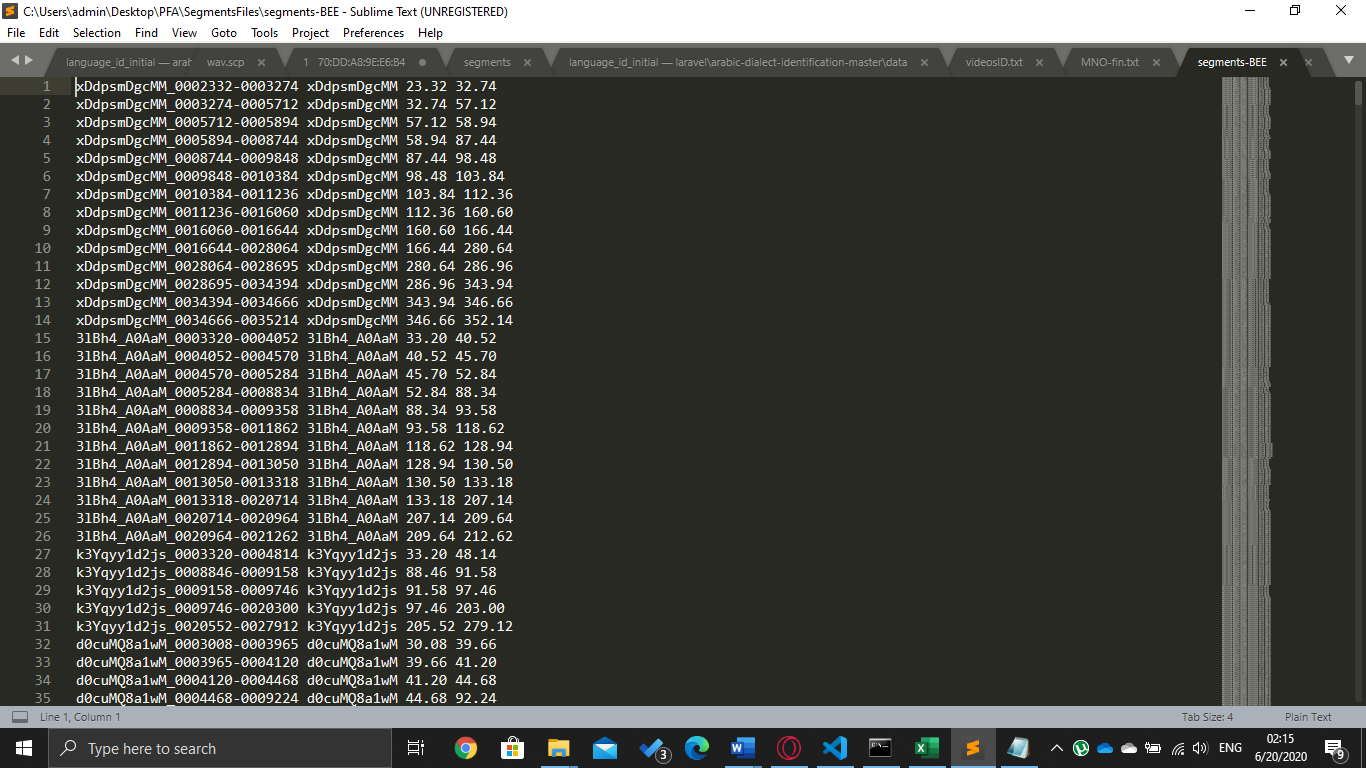


Figure 16 : Exemple du fichier segments

* + 1. La segmentation des audios

Cette partie nous permet de faire la segmentation physique automatique en utilisant le l’outil « Pydub ».

**Pydub ?**

Parfois, dans un script ou une application, nous devons effectuer des tâches de traitement audio. Celles-ci peuvent inclure :

* Chargement et sauvegarde de différents types de fichiers audio.
* La division ou l'ajout d'audio en segments.
* Le mixage de l'audio de deux fichiers audio différents.
* Changer les niveaux audios ou les paramètres de panoramique.
* L'application d'effets simples tels que des filtres.
* Générer des sons audios.

Tout cela peut être réalisé à l'aide de Pydub, un module Python simple et bien conçu pour la manipulation audio. Pydub est mon outil de référence pour les scripts audio de base. Selon les auteurs de PyDub.

A partir des segments obtenus par le script de segmentation théorique et travers un script python qui prend en entrée ce fichier segments et la liste des identifiants initiale on a réussi a extraire les segments de parole de toute les fichiers « Wave ». Et comme dans l’exemple sur la figure ci-dessus. Le script permet aussi de regrouper tous les segments obtenus pour chaque audio d’origine dans un dossier avec le nom de ce dernier en gardant bien sur la même format « WAV ».

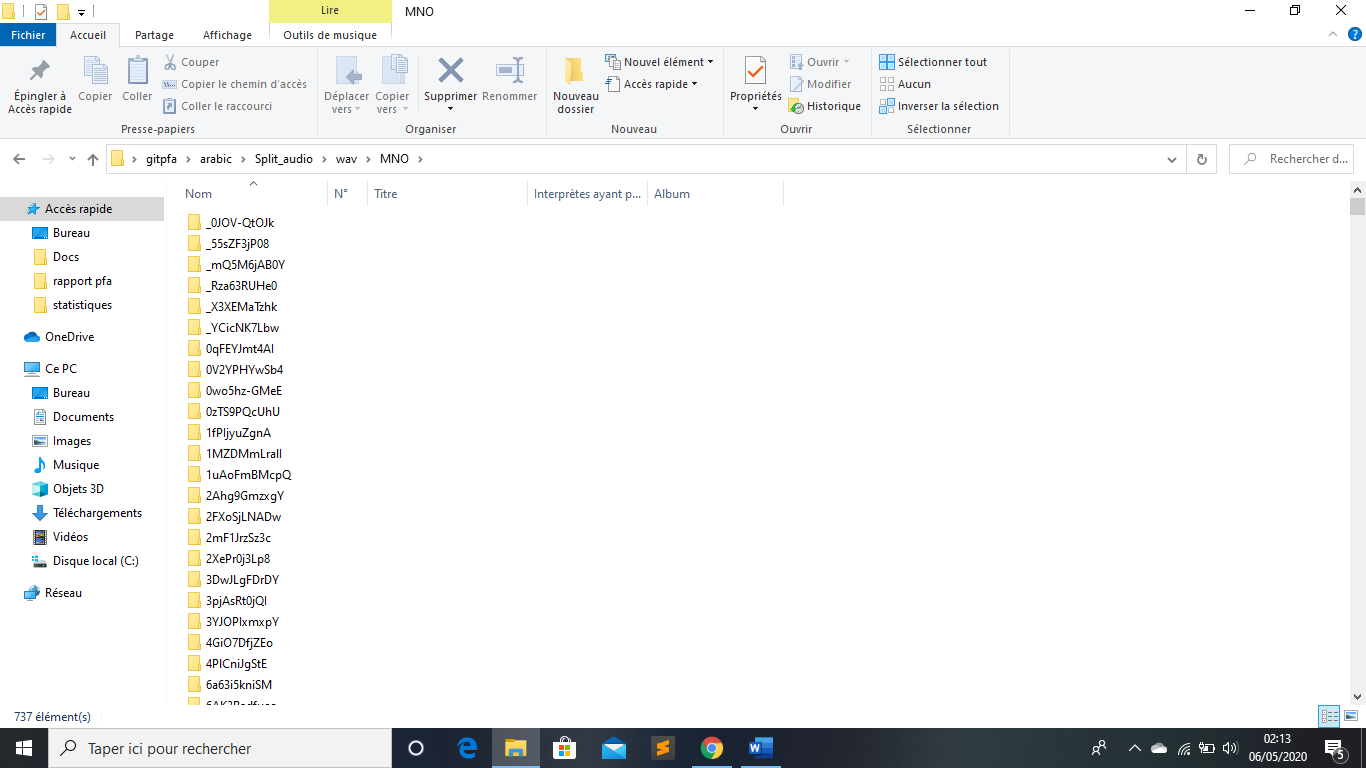
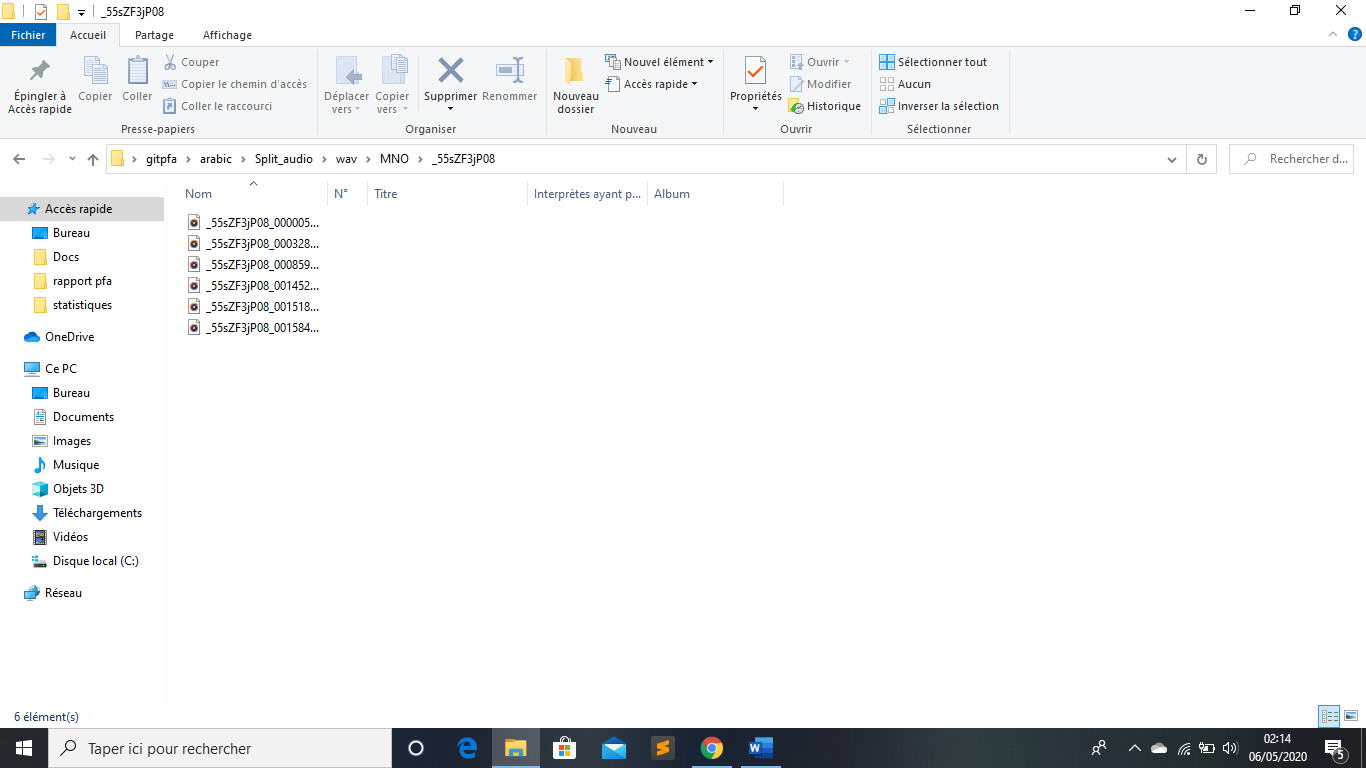
 

Figure 17 : Exemple de segmentation physique

* 1. Evaluation des résultats

Dans cette partie, on va décrire ainsi que donner des graphes représentatifs du nombre d’heures pour chaque dialecte après le crawling. Comme on donnera le nombre d’heures des audios validés et des statistiques contenant le pourcentage des audios supprimés.

1. Nombre d’heures par dialecte

En utilisant les fichiers contenant les ids vidéos, dialectes label, genre et durée générer par le script Info Vidéos, on a utilisé un script pour contrôler le nombre d’heures total qu’on veut utiliser de chaque dialecte, on doit citer qu’on a pu collecter des centaines d’heures d’audios par chaque dialecte.

|  |  |
| --- | --- |
| **Dialectes** | **Nombres d’heures** |
| Urbains | 20h |
| Montagnes du Nord-Ouest | 20h |
| Bédouins de l’Ouest | 20h |
| Bédouins de l’Est | 20h |
| Hassanya | 20h |
| Tarifit | 20h |
| Tachelhit | 20h |

Tableau 2 : Nombre d’heures initial par dialecte

Le nombre d’heures des audios pour chaque dialecte a été choisi après avoir accompli la première étape qui a été la recherche des vidéos d’après le web. Cette dernière nous a permis de distinguer que le dialecte Urbain qui a le plus de ressources mais pour garder les résultats équivalents on a travaillé avec le même nombre d’heures.

Pour le dialecte de tachelhit (TAC) on rencontré un problème au niveau du crawl des vidéos ,lorsqu’on lance le script une erreur d’échec de téléchargement est survenue, on a essayer de trouver d’autres vidéos mais la même chose l’erreur de téléchargement est encore survenue .

1. Nombre d’heures obtenue après la segmentation de chaque dialecte

Ici on va travailler avec un script qui nous donne la différence entre le temps de début et le temps de fin de chaque segment de chaque vidéo, par la suite il va faire la somme des durées obtenue, après on va faire la somme totale des durées des segments initial de chaque dialecte, afin de faire la soustraction des deux résultats pour qu’on puisse remplir les tableaux ci-dessous.

* **Statistiques de parole par chaque dialecte**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dialecte**  **Label** | **Nombre d’heures**  **Initial**  **(h, min, sc**) | **Nombre d’heures supprimée**  **(h, min, s**) | **Nombre d’heures obtenue**  **(h, min, s**) |
| **MNO** | **20:55:09** | **7:36:28** | **13:18:40** |
| **URB** | **20:04:43** | **3:17:01** | **16:47:41** |
| **BEO** | **20:36:36** | **1:23:47** | **19:12:48** |
| **BEE** | **20:47:04** | **7:09:05** | **13:37:58** |
| **TAR** | **20:32:50** | **9:43:43** | **10:49:06** |
| **HAS** | **20:32:56** | **8:05:15** | **12:27:40** |
| **TAC** | **21:09:44** | **11:33:32** | **9:36:11** |

Tableau 3 : Statistiques de parole pour chaque dialecte

Figure 18 : Graphe statistiques de parole

* **Statistiques durée des segments par chaque dialecte**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Dialecte label** | **MNO** | **URB** | **BEO** | **BEE** | **TAR** | **HAS** | **TAC** |
| **Nombre de segments** | **3532** | **5722** | **5997** | **4031** | **6807** | **4345** | **9224** |
| **Durée maximale des segments (sec)** | **751.80** | **531.02** | **487.82** | **579.56** | **146.64** | **148.38** | **84.54** |

Tableau 4 : Statistiques de segments pour chaque dialecte

* **Classification de segments par chaque dialecte**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dialecte label** | **MNO** | **URB** | **BEO** | **BEE** | **TAR** | **HAS** | **TAC** | **Total** |
| **Nombre de segments qui on une durée < 2 sec** | **398** | **1520** | **1484** | **935** | **2192** | **963** | **4239** | **11731** |
| **Nombre de segments qui on une durée entre 2 et 5 sec** | **703** | **1617** | **1616** | **1018** | **2319** | **1182** | **3038** | **11493** |
| **Nombre de segments qui on une durée entre 5 et 20 sec** | **944** | **1856** | **2086** | **1474** | **1982** | **1564** | **1801** | **11707** |
| **Nombre de segments qui on une durée > 20 sec** | **585** | **727** | **811** | **604** | **314** | **636** | **146** | **3823** |

Tableau 5 : Classification de segments pour chaque dialecte

Figure 19 : Graphe statistiques de segments

Figure 20 : Graphe statistiques durée des segments

c. Totale d’heures des audio validés

|  |  |  |
| --- | --- | --- |
|  | **Audios obtenues**  **(h**) | **Audios supprimés**  **(h**) |
| **Tous les dialectes** | **100** | **43** |

Tableau 6 : Totale d’heures des audios supprimés et obtenues

Après avoir accomplir toutes les étapes sur lesquelles la tâche de la segmentation se base, le tableau 5 est une représentation finale montrant le nombre d’heures final des audios obtenue ainsi que ceux supprimés en raison de silence et de bruit.

1. Validation des résultats
   1. Introduction

La nature de notre projet nécessite de disposer d’un grand corpus de parole pour le développement d’un système d’identification automatique de la parole. Ces ressources obtenues sont des données extraites d’après des vidéos sur YouTube en utilisant YouTube data API sur les dialectes de l’arabe marocaine. Par conséquent, une première façon d’accélérer la portabilité des systèmes d’identification de paroles est de développer une méthodologie qui permet une validation rapide des résultats des ressources orales. Pour cela on a organisé notre travaille sur le train, dev et le test en écoutant manuellement les segments qu’on a eu en faisant des tests sur la qualité des audios et l’appartenance des données (paroles des audios) au dialecte cible, on a aussi travaillé sur le test des résultats de la segmentation s’il y on a encore de la musique dans les segments ou s’il y a du silence dans ces audios. Ce traitement manuel des données vise à avoir un large corpus de parole pour la langue maternelle parlée « Darija ».

Cependant, la validation de tels corpus est un peu difficile et va prendre beaucoup de temps pour la réaliser, vu qu'il existe un très grand nombre de données de parole pour chaque dialecte.

5  
Conclusion et Perspectives

Bibliographie

[1] : <https://www.persee.fr/doc/tiers_0040-7356_1974_num_15_59_2031>

[2] : Au Maroc, la reconnaissance constitutionnelle date de juillet 2011, mais sa réhabilitation remonte à 2001 (discours d’Ajdir) à la création de l’IRCAM (2003).

[3] : <https://www.cairn.info/revue-francaise-de-linguistique-appliquee-2007-1-page-71.htm>

[4] : <https://www.cairn.info/revue-francaise-de-linguistique-appliquee-2007-1-page-71.htm>

[5] : Cette partie est prise d’un ancien projet pfa sur les corpus existants .

[6] S. Al-Humoud, N. Al-Twairesh, M. Altuwaijri, et A. Almoammar, « Arabic Spam Detection in Twitter », 2016. <https://www.researchgate.net/Speech-Corpora-for-Arabic-dialects_tbl1_308892800>

[7] « CALLFRIEND Egyptian Arabic - Linguistic Data Consortium ». [En ligne]. Disponible sur : <https://catalog.ldc.upenn.edu/LDC96S49>

[8] « OrienTel ». [En ligne]. Disponible sur : http://www.elra.info/en/projects/archived-projects/orientel/.

[9] « BBN/AUB DARPA Babylon Levantine Arabic Speech and Transcripts - Linguistic Data Consortium ». [En ligne]. Disponible sur : https://catalog.ldc.upenn.edu/LDC2005S08.

[10]N. Y. Habash, Arabic Natural Language Processing. Morgan & Claypool Publishers, 2010.

[11]A. Elmadany, « Automatic Act Classification for Arabic Dialogue Context », 2016.

[12]Y. A. Alotaibi, M. Alghamdi, ET F. Alotaiby, « Speech Recognition System of Arabic Digits based on A Telephony Arabic Corpus », p. 4.

[13]mahaalrabiah, « King Saud University Corpus of Classical Arabic (KSUCCA) », Maha Al-Rabiah - Blog, 20-juill-201

[14]ADI17 : «https://groups.csail.mit.edu/sls/downloads/adi17/index.cgi»

[15] MDB5 : « https://arabicspeech.org/mgb5»

[16] TIDIGITS: « <https://catalog.ldc.upenn.edu/LDC93S10> »

[17] EUROM: « <https://www.hlt.inesc-id.pt/w/EUROM.1_Corpus> »

[18] TIMIT : « <https://catalog.ldc.upenn.edu/LDC93S1>»

[19] BDSONS: « <https://catalogue.elra.info/en-us/repository/browse/ELRA-S0005/> »

[20] ATIS : « <https://catalog.ldc.upenn.edu/LDC95S26>»

[21] BREF: « <http://catalog.elra.info/en-us/repository/browse/ELRA-S0006/> »

[22] WSJ : « <https://catalog.ldc.upenn.edu/LDC2000T43> »

[23] BN : « <https://www.english-corpora.org/bnc/> »

[24] ESTER: « <https://catalogue.elra.info/en-us/repository/browse/ELRA-S0241/> »

[25] CONSOLE GOOGLE CLOUD: « https://console.cloud.google.com »

[26] QAuth2: « https://developers.google.com/identity/protocols/oauth2 »

[27] Google-Auth-Oauthlib / Google-Auth-Httplib2 :

«https://developers.google.com/youtube/v3/quickstart/python »

[28] Categories Youtube : « <https://gist.github.com/dgp/1b24bf2961521bd75d6c> »

[29] [Movavi](https://www.movavi.com/fr/support/how-to/comment-convertir-fichier-audio-en-wav.html) :

« https://www.movavi.com/fr/support/how-to/comment-convertir-fichier-audio-en-wav.html ».

[30] Fréquence d’échantillonnage:

«https://fr.wikipedia.org/wiki/fréquence\_d%27échantillonnage »

[31] CNN: « https://fr.wikipedia.org/wiki/Réseau\_neuronal\_convolutif »