

IDM

VideoGen

Ferey Antoine
2017-2018

I. Configuration

OS Ubuntu 16.04

ffmpeg 3.4.1

VideoGenTools Xtend

BackEnd Java 8, Spring 4.3.10

FrontEnd AngularJS

II. VideoGenTool

1. Comportement

Avant de générer le fichier vidéo, on regarde si il n'y a pas d'erreur d'interprétation :

- probability d'une optional doit être inférieur ou égal à 100 et supérieur ou égal à 0
- la somme des attributs probability dans une alternative doit être égale à 100
- id non redondant
- le fichier n'existe pas
- si le paramètre duration n'est pas supérieur à la durée de la vidéo
- la couleur du texte doit être valide (vu qu'elle n'est pas définie dans la grammaire)
- id obligatoire (contraire à la grammaire mais très utiles dans la suite)
- le texte doit être un mot (un problème que j'ai pas pus résoudre)

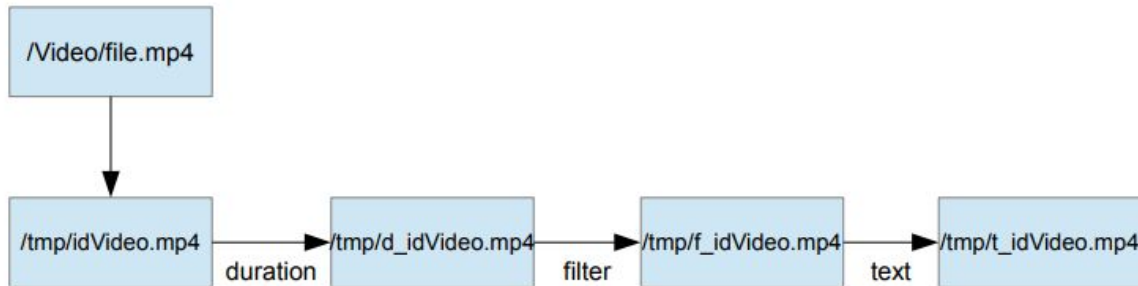
Si il n'y a pas de probability présente dans le cas d'une vidéo optionnal, elle sera de 50%. Dans le cas de videos alternative, elles seront équiprobables.

2. Modification des vidéos

Toutes les commandes ffmpeg sont définie dans le fichier Util.xtend.

La première étape est de copier les vidéos dans le dossier tmp avec le nom de son id dans le fichier videogen. Avec cela on réalise les changements sur des copies et non sur l'original. On les enregistre avec le nom de l'id (et non avec le nom du fichier) car il peut arriver que dans une spécification videogen on peut utiliser plusieurs fois la même vidéo.

Pour chaque changement (duration, filter, ajout d'un texte) on crée une nouvelle vidéo dans le dossier *tmp* puis on modifie la location du media.



On peut voir qu'il y a des préfixes pour chaque changement.

Après tous ces traitements on peut se retrouver avec un fichier ffmpeg écrit dans ce style :

file '/tmp/t_id1.mp4'

file '/tmp/id2.mp4'

file '/tmp/f_id31.mp4'

file '/tmp/d_id5.mp4'

Tous les fichiers ffmpeg sont créés dans le dossier *tmp* avec comme nom MMdd_HHmmss_SSS.ffmpeg.

3. Problèmes rencontrés

Les principaux problèmes rencontrés sont liés à l'utilisation de la concaténation de ffmpeg. Même si le format vidéo était le même (mp4,avi,...) , ffmpeg pouvait être en erreur.

Du coup toutes mes vidéos possèdent les mêmes caractéristiques pour éviter de rencontrer des erreurs :

- format : mp4
- largeur : 1024
- hauteur : 576
- fréquence : 25 trames/s
- audio : AC3

J'ai utilisé le logiciel **OpenShot VideoEditor** pour extraire les vidéos sous ce même format.

Un autre problème était liée à l'encodage audio. ffmpeg modifie l'encodage audio après l'application d'une modification. Lors de la lecture de la vidéo transformée il y avait bien du son. Mais lors de la concaténation, le son n'était plus présent sur les vidéos transformées.

Pour résoudre ce problème, lors de la modification, on force l'encodage dans le format initial (ici du AC3).

Cas particulier sur la grammaire rencontré :

optionnal videoseq v1 "path/to/file.mp4"

optionnal videoseq v1 "path/to/file.mp4" { probability 0 }

Dans le xtend avec la condition (video.probability == 0), les deux ci-dessus remplissent la condition alors qu'il ne disent pas la même chose.

Il n'y a donc pas de manière de les différencier. Dans mon projet je considère que { probability 0 } veut dire 50% de chance d'apparaître.

Je ne me suis pas occupé des images. Vu que la concaténation entre vidéo représente déjà un challenge, ajouter des images aurait été sans doute très compliqué.

III. Site Web

1. Concepte

Pour le site web, plutôt que d'avoir une variante sur une unique spécification videogen, j'ai pris la décision de créer quelque chose de plus concret (comme bref, canal+ à 30 ans). Les possibilités sont tellement vaste avec videoGen que je prend le risque de changer cette partie du projet.

Le site web génère un quizz vidéo sur des films en fonction du choix de l'utilisateur. Premièrement, l'utilisateur choisit un acteur. Il arrive ensuite sur une liste de films et doit en choisir 3 films pour créer le quizz.

La vidéo générée est composée :

- une intro avec le nom de l'acteur choisie
- un extrait d'un film avec l'acteur
- une transition de 4 sec
- la réponse (le moment du film ou le nom du film est présent)
- deux autres (extrait → transition → réponse)

Les extraits d'un film avec l'acteur sont des alternatives. C'est à dire pour la même requête (acteur A, film 1, film 2, film 3) on peut se retrouver avec un quizz différent.

2. Architecture

BDD :

Contient les informations sur les films, les acteurs et la localisation des extraits des films.

BackEnd :

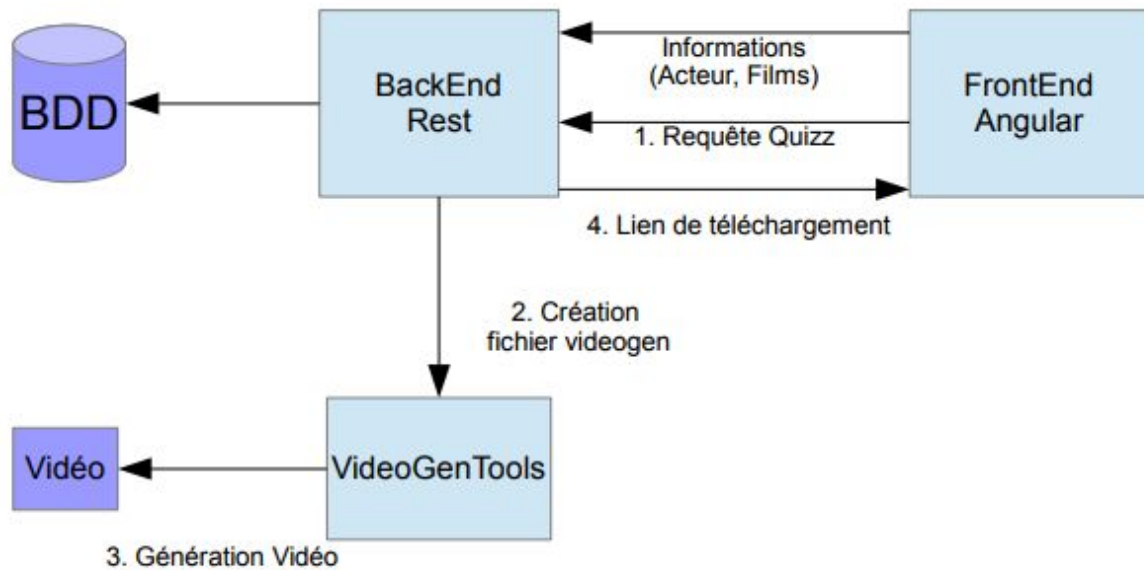
Renvoie les informations de la BDD. Créer un fichier videogen depuis une requête du serveur FrontEnd et retourne la vidéo générée par la librairie VideoGenTools.

Extraits Vidéos :

Les vidéos sont présentes coté BackEnd.

FrontEnd :

Réaliser très simplement via AngularJs.



3. Mise en place

BackEnd

Accessible depuis le port 8080

Répertoire du projet BackEnd

git clone <https://github.com/Kwodhan/VideoGenBackEnd>

Important : On met le dossier "Video" dans le répertoire home de l'utilisateur (~). Le dossier "Video" est présent dans le dossier du répertoire git.

Lancement de la BDD HSqI

`./run-hsqldb-server.sh`

Lancement du serveur

`./java -jar server-1.0.jar`

FrontEnd

Accessible depuis le port 9000

Répertoire du projet FrontEnd

git clone <https://github.com/Kwodhan/VideoGenFrontEnd>

Lancement du serveur

`grunt serve`

4. Problèmes rencontrés

Trouver des bon extraits d'environ 1 minutes depuis un film est une tâche assez longue. C'est pour cela qu'il y a seulement deux acteurs et 4 films pour chaque.

Mes coupes ne sont pas parfaites pour certains extraits et les transitions ne sont pas très élégante.

Les quizz dure entre 3 et 4 minutes. La Gification des vidéos n'est pas intéressante pour le site web. La fonctionnalité existe tout de même.

La création de la vidéo prend plusieurs dizaine de secondes, ce qui représente un certains temps quand on navigue sur le web.

J'ai essayé d'utiliser docker pour une mise en production rapide mais il y a eu encore un problème avec ffmpeg qui concatène mal les vidéos depuis un docker. Il y a bien le son mais plus les images.

IV. Etude empirique

1. Production

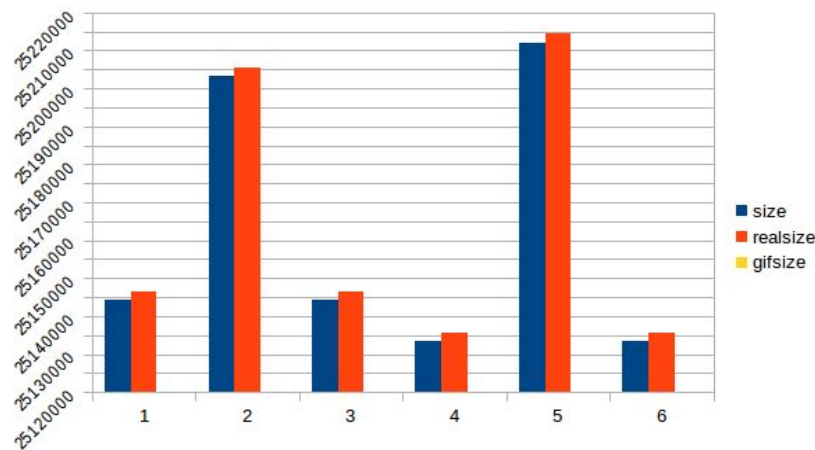
La production des fichiers csv se fait via la class VideoGenEtude. J'ai étudié les données grâce à Excel car je ne connais pas suffisamment le R pour réaliser des statistiques.

J'ai une spécification qui comporte 2 mandatory, 1 alternative et 1 optional. Il y a trois exemplaires :

- Sans Changement
- Filtre Noir et Blanc
- Filtre Negate

Ici on va parler de taille réel (fichier final généré) et de taille théorique (somme des extraits).

2. Sans Changement



Nous avons ici la taille par fichier pour chaque variantes (taille total / nombre de vidéo).

On peut voir pour les 6 variantes que à chaque fois la taille réel est plus élevé que la taille théorique. La différence entre taille réel et taille théorique est d'environ 2300 octets. On peut se dire que cela représente les métadonnées mais après une commande ffmpeg :

```
ffmpeg -i resultat.mp4 -f ffmetadata metadata.txt
```

On peut voir que le fichier fait seulement 200 octets environ. Cette ajout reste donc un mystère.

3. Filtre

Pour les deux filtres (b&w et negate), je fus surpris. Dans certains cas, la taille théorique est plus grande que la taille réel. Au premier regard, je me disais que l'algorithme prenais la taille des extraits vidéo avant la transformation. Mais après vérification (présence du préfixe sur les fichiers analysés, voir Modification des vidéos) il n'y avait pas d'erreur.

4. Gif

Pour les Gifs il n'y a pas de chose très intéressante à dire. Les Gif prennent moins de place que les fichiers vidéo. Pour les filtres, le filtre b&w prend moins de place que sans les modifications et le filtre negate prend plus de place que sans les modifications.

5. Conclusion de l'étude

Les fichiers excel sont fournis dans le dossier Etude. Je ne vois pas de corrélation entre les filtres et les tailles. Si j'avais pris le temps d'apprendre des outils statistiques tel que le R, j'aurais sans doute trouvé une corrélation.

V. Conclusion

Ce projet représente une grande quantité de travail. C'est peut être lié que je sois seul sur le projet. Je n'ai pas pu réaliser autant de chose que je voulais (surtout au niveau site web).

A propos du site web, moi qui suis un grand amateur de cinéma, pouvoir avoir une plateforme pour réaliser des quizz était vraiment sympas. Choisir les bon extraits, différents acteurs était certe long mais très agréables. Il pourrait y avoir plein de nouvelles fonctionnalités :

- extrait en fonction d'un réalisateur
- niveau de difficulté
- un quizz random

Pour finir, j'ai l'impression d'être devenue plus un expert ffmpeg qu'en DSL. J'ai passé plus de temps sur les commandes ffmpeg et sur le site que sur le Xtend. Cela en résulte que je n'ai pas travaillé sur la partie concours.