Suggestions d'évolutions fonctionnelles de documentare

Joël Gardes

6 mars 2017

Les modes de fonctionnement de documentare

Actuellement, la bibliothèque documentare prévoit deux modes de fonctionnement : scrutation de répertoires, lecture de json. Ces deux modes fonctionnent et donnent d'assez bons résultats. Le mode répertoire permet de traiter des volumes importants de données. Le mode json est nettement plus "intégré" en ce sens qui permet de construire un document contenant l'ensemble des données utiles au scénario de traitement.

C'est ce mode json qu'il conviendra de faire évoluer en ce sens que les données utiles doivent à termes devenir l'ensemble des informations nécessaires à un traitement, que l'on travaille sur une liste de documents, une liste de fragments issus de plusieurs documents (p. ex. les logos que l'on aura extrait d'outils tiers), une liste de blocs correspondant à la structure d'un document (titres, paragraphes, etc...) qui sont à soumettre à une transcription (OCR) ou un reflow.

Ainsi, le json construit par "prep-data", doit évoluer vers un format pivot, dont le modèle est inféodé au métier à l'origine des intentions de traitement.

Dans le cas spécifique de la classification de documents dans un corpus déjà trié, cela reviendra à disposer d'un json reflétant la base de documents triés et d'un json du document à classifier dont la structure sera identique au modèle des documents classés. Dans le json descriptif de la base de documents les références aux contenus devront tenir compte d'éventuels multisets correspondant aux modèles représentifs d'un document. La même logique se fera dans le cas de l'OCR où la granularité du json du document à traiter est au niveau du glyphe : le json du document d'entrée sera composé de la liste des glyphes et sera comparé par similarité aux différents multisets représentant la base d'OCR.

Les formats de mesure de la similarité

Cette partie a une dimension prospective car elle conduit à la définition d'un format pivot compatible avec une interface comme "Zénobie" et un service de mesure de similarité. Elle correspond donc à la définition d'une chaîne documentaire s'inscrivant dans un travail de recherche sur le document numérique.

Toutefois, il pourrait être très valorisant de commencer à appliquer certains principes dans la constitution du

json produit par "prep-data"

Images

Nous avons retenu la mesure de similarité sur les bitmaps (un format raw "propriétaire" incluant un balisage de séparation de chaque contenu mesuré).

Cette approche permet de s'abstenir des contraintes liées aux formats telles que les problèmes liés aux compressions JPEG par exemple, ou liés au codage de l'image en PNG. La séquence raw produite est linéaire et sa trame rend compte du début du contenu et des passages à la ligne suivante. Il apparaît, dans la mesure, que l'utilisation de balises "hors alphabet de codage du contenu", serait intéressante. En effet, actuellement, le délimiteur de contenu est un string ("JoTophe") dont la probabilité de survenue dans la séquence d'octet est faible, voire négligeable, il en est de même pour le délimiteur de ligne "\n".

Une solution est possible en passant par une "compression d'alphabet" comme celle que nous offre un transcodage en Base64. Ce transcodage est totalement réversible en ce sens qu'il n'occasionne aucune perte d'information. Il permet de passer d'un alphabet de 128 octets à un alphabet de 64 octets.

Un tel transcodage augmente la taille des données inscrite dans la séquence d'octets mais préserve l'information du contenu. Son avantage serait ici de "libérer" 64 octets qui deviendraient dédiés au balisage des séquences d'octets à traiter (en reprenant, pour commencer, le balisage actuel). Si, pour le traitement d'images, nous n'avons pas encore d'idées sur l'utilisation de ces "nouvelles balises" potentielles, il n'en est pas de même pour le traitement d'autres contenus.

Textes

Actuellement, le traitement porte sur le texte tel quel et comporte les même limitations que dans le cas de l'image en ce qui concerne la probabilité de survenue du balisage de la trame de contenu. Un transcodage en Base64 autoriserait la même libération d'un jeu de balises qui, ici, pourrait servir à inclure des métadonnées sur la nature des substrings contenus dans un texte, comme par exemple, des entités nommées (noms propres, syntagmes, différenciation tiret/césure...), ce qui permettrait d'ouvrir le champs à des traitements incluant une dose de sémantique sur le texte.

Graphiques

Le problème du graphique (SVG, DWG, DXF...) est que son séquencement n'est pas liée à une grille comme dans le cas d'une image ou d'un texte et, en particulier dans le cas de tracés, ce séquencement fait appel à un vocabulaire ouvert, à savoir, les coordonnées de points. Cela signifie que les mesures de similarité ne permettent que de détecter des motifs parfaitement alignés sur une grille connue. Il s'agira, à terme, d'étudier l'intérêt d'une rasterisation des tracés graphiques.

Le json produit par "prep-data"

Ce json doit devenir le document "métier" des traitements incluant une mesure de similarité et couplé avec les usages des contenus choisis en entrée. En ce sens qu'il devra, à terme, être capable de décrire :

- la granularité de l'information à traiter : une liste de documents, de fragments produits par segmentation, etc.
- la nature du contenu à traiter : image, texte, graphismes, fichiers bruts, etc.
- dans le cas de fragments de contenus : leur localisation dans le document source (le document lui-même et les coordonnées de la zone d'intérêt contenante)
- dans le cas de textes ou de fragments de textes : la référence à un modèle (pour, par exemple, les entités nommées)
- La fonction du document (par exemple, les multisets d'une base de référence pour les OCR)

Le format de codage des segments de contenus devra progressivement privilégier le Base64 avec, pour l'image, une conversion raw qui, si on généralise le principe, devriendra : une linéarisation des contenus avant mesure.

Ce json pivot, selon la taille des données présentes, contiendra soit les segments eux-même, soit le lien vers les fichiers (qui devront avoir le même format que les segments).

Ce json devra, en outre, pouvoir être lu par une interface spécifique capable de reconstruire une vue du ou des documents sources (donc, soit un visualiseur du document, soit un "gopher" affichant les documents sources et leur localisation).

Le modèle de ce json reprend l'approche hiérarchique du modèle de document construit dans ozalid en étendant son emprise. Par exemple :

- une page de document est un fichier appartenant à une arborescence qui est le livre auquel appartient cette page.
- un billet de train est un fichier appartenant à un dossier contenant l'ensemble des billets de train.
- un prospectus produit est un fichier destiné à remplir un classeur de produits.

Le fait de commencer à être capable d'inclure le document "métier" auquel doit appartenir le document à traiter permettra d'ouvrir la voie à la personnalisation des traitements et aux actions de décisions sur les contenus : "ces documents sont des billets de train, j'en fait quoi et je les mets où?"