Université de Nantes
Licence 3 Informatique

Alignement multimodal de contenus éducatifs

Un stage de recherche en Traitement Automatique du Langage Naturel

10 juin 2014

Réalisé par : Matthieu Riou

Encadrants : Colin de La Higuera, Solen Quiniou, Olivier Aubert
Encadrante Universitaire : Irena Rusu

Stage effectué du 14 Avril au 11 Juillet 2014
au Laboratoire Informatique de Nantes Atlantique

Remerciements

Je tiens à remercier mes encadrants, pour leur sympathie et leurs conseils, et pour m'avoir beaucoup appris.

Merci également au personnel du LINA, pour leur disponibilité et leur accueil.

Je remercie aussi les stagiaires et les doctorants qui ont permis que ce stage se déroule dans une ambiance géniale.

Table des matières

1 Introduction

1.1 Contexte

Le développement des MOOCs (Massive Open Online Course), en France et à l'étranger, est en train de changer notre façon d'appréhender l'apprentissage. Beaucoup de ressources éducatives libres sont produites chaque jour, mais leur accès et leur traitement restent complexes. Afin de faciliter la navigation entre ces ressources, ainsi que leur utilisation, nous avons essayé de comparer ces ressources de natures diverses.

Ce stage porte sur l'alignement multimodal. Aligner deux documents signifie faire correspondre ensemble des parties de ces deux documents, et le terme multimodal indique que l'alignement se fait entre plusieurs formats, ici une vidéo et un document textuel.

Ce rapport va vous présenter nos pistes de réflexion, les différents résultats que nous avons obtenus et ce qu'ils signifient.

1.2 Présentation de l'équipe

Ce stage se déroule au Laboratoire d'Informatique de Nantes Atlantique (LINA) de Nantes pendant trois mois dans l'équipe Traitement Automatique du Langage Naturel (TALN). Au moment de la rédaction de ce rapport, il reste encore un mois de stage.

Le LINA est composé de 9 équipes, pour près de 180 personnes. L'équipe TALN comprend 11 membres permanents et 6 membres non-permanents. Ses thèmes de recherche sont regroupés en deux groupes. D'une part, l'analyse et la découverte, qui s'intéressent à l'analyse des langages naturels et à la production de ressources linguistiques comme des corpus. D'autre part, l'alignement et la comparaison de documents, possiblement multilingues ou multimodaux (plusieurs formats : texte, son, vidéo...). C'est dans ce domaine que ce stage se positionne.

Je suis encadré par Colin de la Higuera et Solen Quiniou, enseignants-chercheurs de l'équipe TALN, et par Olivier Aubert, ingénieur de recherche pour la plateforme COCo (CominOpenCourseware).

Mon encadrante universitaire est Irena Rusu de l'équipe Combinatoire et Bio-Informatique (ComBi).

1.3 Objectif

Le but de ce stage est de pouvoir aligner automatiquement une vidéo et un document textuel, c'est-à-dire de pouvoir lier de manière automatique

chaque passage de la vidéo avec les morceaux de texte correspondant au sujet abordé, puis d'arriver à trouver un alignement global sur la vidéo entière.

Pour travailler, nous utilisons actuellement des articles scientifiques et des vidéos de conférences présentant ces articles, donnés par le projet `translectures` [1].

Le stage aborde autant les problématiques mathématiques et algorithmiques que celles de visualisation et d'évaluation des résultats.

1.4 Environnement du stage

Le projet est rattaché à la plateforme `CominOpenCourseware` [2] (COCo), qui est la plateforme de ressources pédagogiques ouvertes de l'Université de Nantes.

La plateforme COCo a pour but de produire des contenus ouverts multimodaux, et c'est naturellement que s'est posé pour elle le problème d'alignement sur lequel nous travaillons.

1.5 Plan du rapport

Dans ce rapport, nous décrirons d'abord les tâches à réaliser, ainsi que l'état de l'art actuel sur le problème d'alignement multimodal. Nous présenterons ensuite les données à notre disposition, puis nous expliquerons le travail réalisé, en détaillant les approches et les méthodologies. Enfin nous discuterons des résultats, en décrivant les différentes pistes de réflexion possible pour la suite du stage.

1. `www.translectures.eu`
2. `www.comin-ocw.org`

2 Description du projet

2.1 L'état de l'art

Les problématiques d'alignement en informatique ne sont pas nouvelles. Les informaticiens et les linguistes ont vite souhaité faire correspondre des textes ou des segments de textes, et ce pour différentes problématiques, comme la fouille d'information ou la détection de plagiat. De même, les bio-informaticiens travaillent depuis longtemps sur les alignements de séquences (sur l'adn ou les protéines).

Le traitement du langage naturel a apporté de nouvelles problématiques d'alignement, en essayant de comparer différents textes. Ce stage porte plus particulièrement sur l'alignement multimodal, c'est-à-dire entre différents formats, ici sur un texte et une vidéo. Pour cela, nous travaillons avec la transcription de la vidéo.

L'alignement de textes se base sur l'identification de segments dans ces textes, puis sur des mesures de similarité entre ses segments. Pour cela, les segments peuvent être représentés comme des vecteurs de termes ou de mots-clés. Des scores leurs sont attribués grâce au tf-idf, qui est classiquement utilisé dans le domaine de la fouille d'informations. Il est en plus possible de prendre en compte des informations linguistiques sur les mots analysés.

L'étape finale est de rechercher les ressemblances entre les différents segments, en fonction de leur score. Cela peut être fait en définissant une mesure de similarité, en utilisant par exemple une distance cosinus, ou en appliquant des algorithmes d'apprentissage ou des méthodes statistiques comme le clustering.

L'aspect multimodal nous oblige à avoir un système robuste, qui pourra alors résister aux erreurs de transcription de la vidéo.

2.2 Les tâches à accomplir

Le premier travail est d'identifier les segments de la vidéo et de l'article, et donc de les découper en morceaux cohérents. Pour la vidéo, le découpage est calqué sur les slides. L'article, quant à lui, est coupé par paragraphes.

Le processus d'alignement comprend deux parties. Tout d'abord il faut définir les mesures de similarité entre la vidéo et l'article, c'est-à-dire donner une valeur qui définit la similarité de deux documents. Autrement dit, trouver quels passages de la vidéo abordent le même sujet qu'un paragraphe de l'article, et dans quelles proportions.

Une fois ces mesures de similarité calculées, nous sommes capables de trouver, pour un passage de la vidéo, le ou les paragraphes lui correspondant le mieux.

Ensuite, nous essayerons de trouver un alignement global, c'est-à-dire, à partir des correspondances entre des passages de la vidéo et des paragraphes de l'article, faire correspondre de manière plus large des blocs de vidéo (plusieurs morceaux) avec des parties de l'article (plusieurs paragraphes). Nous voulons ainsi trouver un fil directeur, dans la vidéo et l'article, et suivre parallèlement le déroulement de l'exposé dans les deux documents.

Enfin, il est aussi nécessaire de mettre au point des interfaces de visualisation, afin de vérifier les données, et d'y naviguer de façon ergonomique, ce pour deux raisons. La première est de nous permettre d'évaluer simplement et rapidement les résultats obtenus, afin de les améliorer. La deuxième est de fournir une interface utilisateur s'appuyant sur nos alignements, et permettant d'explorer plus facilement des contenus éducatifs.

3 Les données à disposition

3.1 Les conférences de Translectures

Translectures est un projet européen visant à développer des solutions efficaces pour la transcription et la traduction automatiques de contenus éducatifs en ligne [3].

Translectures met à disposition de nombreuses conférences. Nous avons ainsi accès à environ 8000 conférences, la plupart en anglais. Pour chaque conférence, nous disposons de sa vidéo, de son article scientifique en pdf et de ses slides en pdf.

De plus, nous disposons de fichiers plus spécifiques contenant des informations sur la vidéo.

3.1.1 Transcription de la vidéo

Nous avons à disposition une transcription de la vidéo, en format standard TTML [4] (anciennement dfxp). Un exemple de fichier est disponible dans l'annexe A.1 page 27.

Le fichier peut être analysé comme un fichier xml. La balise *header* nous donne plusieurs informations intéressantes. Par exemple, elle nous indique si la transcription est humaine ou automatique (attribut `aT`), quel extracteur l'a effectuée (attribut `aI`) ou les temps de début et de fin (les attributs `b` pour begin et `e` pour end).

La transcription est découpée en segments (de façon arbitraire, pour le sous-titrage de translecture) grâce aux balises *tl :s*, indexées par l'attribut `sI`, avec un temps de début et de fin (les attributs `b` et `e`).

```
<tl:s sI="0" cM="0.7800" b="0.00" e="25.02">
```

Chaque balise *tl :w* représente un mot. Elle indique le temps où il a été prononcé (les attributs `b` et `e`), ainsi qu'un indice de confiance (l'attribut `cM`), compris entre 0 et 1, nous indiquant la probabilité du mot d'être le bon.

```
<tl:w cM="1.0000" b="0.00" e="0.02">~SILENCE~</tl:w>
```

Parfois, le mot donné par la balise *tl :w* est ~SILENCE~ (ou [SILENCE] selon les transcriptions). Il existe donc deux façons de trouver du silence dans la transcription. Dans le cas d'une balise ~SILENCE~, ou quand il y a du temps entre la fin d'un mot et le début du suivant.

3. www.translectures.eu/web/about/
4. www.w3.org/TR/ttaf1-dfxp/

3.1.2 Transcription des slides

Nous avons aussi accès à une transcription des slides, encore au format TTML. Un exemple de fichier est disponible dans l'annexe A.2 page 28.

Nous ne nous sommes pas encore intéressés à la transcription des slides, même si elle pourrait nous offrir des informations intéressantes. Ce fichier nous a en revanche donné, pour chaque slide, son temps de début et de fin.

3.2 Travaux précédents

Plusieurs personnes avaient déjà travaillé sur chaque partie du sujet, nous avions donc déjà, dès le départ, des travaux sur lesquels nous baser.

3.2.1 L'interface

Clément Horhant, Benjamin Le Clere, Antoine Leboeuf, Titouan Pasquet et Léo Turrado, en deuxième année de l'IUT de Nantes, ont travaillé avec Solen Quiniou sur une interface web.

Leur objectif était de réaliser une application web permettant de synchroniser un fichier pdf et un fichier vidéo. Le but était de pouvoir naviguer entre les deux médias, et d'afficher un alignement. Le travail restant étant d'arriver à construire automatiquement cet alignement.

Ainsi, lorsque l'alignement est fourni, au fur et à mesure du déroulement de la vidéo, les paragraphes correspondants de l'article sont encadrés de façon synchronisée. De plus, cliquer sur un paragraphe du pdf nous amène directement au moment correspondant dans la vidéo.

Il est aussi possible de créer manuellement l'alignement. Il faut passer en mode édition, il est alors possible de double-cliquer sur un paragraphe, et de sélectionner pour ce paragraphe les temps de début et de fin du passage correspondant dans la vidéo.

Pour réaliser leur interface, ils ont dû trouver les coordonnées des paragraphes dans le fichier pdf. Pour extraire les paragraphes du pdf, ils ont utilisé tika [5], une bibliothèque en java, mais elle ne leur donnait pas les coordonnées. Le programme pdftohtml [6] leur donnait un fichier en xml avec les coordonnées du texte ligne par ligne. Ils ont donc fait correspondre le texte extrait par tika et celui extrait par pdftohtml pour trouver les coordonnées des paragraphes.

Ils ont eu plusieurs problèmes, notamment à cause d'une gestion différente de certains caractères par les deux bibliothèques. Leur solution n'était

5. `http://tika.apache.org`
6. `http://pdftohtml.sourceforge.net`

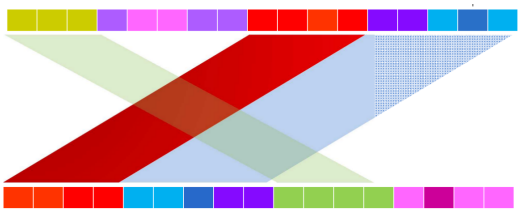
donc pas parfaite, et ils étaient dans l'obligation d'utiliser deux outils différents pour déterminer les paragraphes. Nous nous sommes donc demandé s'il était possible d'obtenir des résultats équivalents en ne gardant qu'un seul outil. Le premier objectif de ce stage a donc été d'extraire les paragraphes et leur coordonnées en utilisant seulement pdftohtml, en se basant sur la typographie.

3.2.2 Les mesures de similarité

Davy Auffret a travaillé sur les mesures de similarité lors du module d'initiation à la recherche en M1 Atal. Son but était de trouver les mesures de similarité entre des passages de la vidéo et des paragraphes de l'article. Nous nous sommes basé sur son travail pour calculer nos propres mesures.

3.2.3 L'alignement global

Colin de la Higuera a travaillé sur la recherche d'un alignement global à partir des distances (l'inverse de la similarité) entre des segments de textes différents. Le principe est d'arriver à faire correspondre des ensembles de segments les plus gros possible. Ainsi, dans le cadre d'une conférence, on remarquera que dans la vidéo, le conférencier a inversé l'ordre de ses parties 2 et 3 par rapport au pdf, et on pourra les aligner correctement.



Un alignement de blocs
Auteur : Colin de La Higuera

Pour aligner un document X découpé en segments $\{x_1, ..., x_n\}$ et un document Y découpé en segments $\{y_1, ..., y_m\}$, il construit un vecteur à 4 dimensions $T[i, j, k, l]$ qui contient les coûts d'alignement du bloc $(x_i...x_j)$ avec le bloc $(y_k...y_l)$, c'est-à-dire la distance entre ces deux blocs.

Il va alors chercher dans ce vecteur la paire de blocs la plus grande ayant le plus faible coût d'alignement. Ces deux blocs seront alors alignés ensemble, et ne pourront plus être alignés par la suite. Cet algorithme va ensuite être répété jusqu'au résultat final.

Actuellement, l'algorithme existe et une implémentation a été écrite en php. Malheureusement il a une complexité en $O(n^4)$, car on doit parcourir

en continue un vecteur à 4 dimensions. Cette complexité est trop élevée pour des applications pratiques. Nous allons donc devoir réfléchir à l'amélioration de cet algorithme.

3.3 Données de l'article

Le programme pdftohtml a été utilisé pour extraire le texte du document pdf. Un exemple de fichier obtenu avec pdftohtml est présenté dans l'annexe A.3 page 29.

Le texte est découpé par page, chacune étant numérotée. Nous connaissons aussi leur taille (attributs `height` et `width`). La première page nous donne aussi des informations sur les polices utilisées (la balise *fontspec*).

```
<page number="1" position="absolute" top="0" left="0" height=
   "1188" width="918">
```

Enfin dans chaque page, le texte est découpé par ligne dans les balises *text*. Le découpage se fait aussi en fonction de la police utilisée. Ainsi, si plusieurs polices sont utilisées sur la même ligne (par exemple pour un exposant, ou une formule mathématique), la ligne sera découpée en plusieurs balises *text*.

```
<text top="101" left="149" width="616" height="37" font="0">
   Automatic Labeling of Multinomial Topic Models</text>
```

Chaque balise text nous donne comme informations les coordonnées haut et gauche du texte qu'elle contient (attributs `top` et `left`), ainsi que sa largeur et sa hauteur (attributs `height` et `width`). Nous avons aussi sa police, identifié par son id (attribut `font`).

4 Présentation du travail réalisé

4.1 Découpage de la vidéo et de l'article

Un premier objectif a été de trouver un découpage cohérent pour la vidéo et l'article scientifique. Pour visualiser les résultats, nous avons utilisé l'interface web développée par le groupe de l'IUT, car elle était déjà prête et fonctionnelle.

4.1.1 Découpage de l'article

Pour l'article, la bibliothèque tika découpait déjà le texte du pdf en paragraphes, et le travail du groupe de l'IUT utilisait en plus pdftohtml pour trouver les coordonnées du paragraphe.

Notre but était d'utiliser seulement pdftohtml qui extrait le texte du pdf ligne par ligne, avec les coordonnées des lignes. Ainsi, on n'utilise plus tika et on ne garde qu'un seul outil, par soucis de simplicité.

Nous nous sommes intéressés à la représentation typographique d'un paragraphe. Le but était de retrouver les paragraphes sans comprendre le texte lui-même, mais seulement à partir des informations que j'avais sur les coordonnées des lignes : position, taille, police...

Pour cela, nous avons écrit un script python qui commence par analyser le document, en supposant que les paragraphes du document représentent la plus grande majorité du texte. La police des paragraphes peut alors être définie comme étant la police la plus présente. De même pour la marge à gauche du paragraphe. Le script peut aussi trouver la marge à gauche d'un alinéa, qui est la marge à gauche la plus présente pour les lignes dont les trois lignes suivantes ont la marge à gauche d'un paragraphe.

Il effectue ensuite un pré-traitement qui consiste à supprimer les lignes considérées comme inutiles. Elles ne seront alors pas prises en compte pour définir les paragraphes. Il supprime les lignes ne possédant pas la police des paragraphes, et les lignes n'ayant pas la marge à gauche d'un paragraphe ou d'un alinéa.

Le pré-traitement réunit aussi les lignes qui ont été mal découpées par pdftohtml, le plus souvent à cause d'un exposant ou d'un indice, ou encore d'une formule mathématique.

Enfin, le texte traité peut être découpé en paragraphes, en délimitant le début d'un nouveau paragraphe à chaque alinéa, et à chaque fois qu'une interligne est trop grande. Dans les cas des articles à plusieurs colonnes, le script vérifie aussi le changement de colonne.

Toutes les options de pré-traitement et de délimitation des paragraphes sont désactivables pour affiner les résultats.

Les résultats sont stockés dans deux fichiers différents. Le premier est formaté pour l'interface web, et donne, page par page, les coordonnées des paragraphes dans des balises *text*. Un attribut `time` est aussi prévu pour stocker les résultats de l'alignement.

Le deuxième fichier contient les résultats dont nous aurons besoin pour l'alignement. Un exemple de ce fichier est présenté dans l'annexe A.4 page 30. Nous donnons à chaque paragraphe un identifiant, et nous récupérons son texte.

4.1.2 Découpage de la vidéo

Pour la vidéo, un découpage sur les temps des slides a été choisi. Nous avons donc écrit un script python qui analyse le fichier de la transcription des slides et celui de la transcription de la vidéo.

Tous les mots prononcés sont regroupés, sans les silences, selon les temps des slides, puis sont stockés dans un fichier xml. Un exemple de ce fichier est présent dans l'annexe A.5 page 31.

Pour chaque passage de la vidéo, nous indiquons ses temps de début et de fin. Nous lui donnons aussi un identifiant (l'attribut id) et nous indiquons à quel slide il correspond (l'attribut sI), les slides étant numérotés à partir de 1.

Un passage peut ne correspondre à aucun slide, par exemple en début de vidéo, si le premier slide démarre plus tard. Dans ce cas, sI vaut 0. Plusieurs morceaux peuvent donc avoir sI qui vaut 0, mais id permet toujours de les identifier.

Afin d'améliorer le découpage, nous plaçons les temps de début et de fin au moment du plus long silence trouvé autour des temps du slide. Cela permet d'avoir moins de chance de découper au milieu d'une phrase.

4.2 Les mesures de similarité

Nous avons ensuite travaillés sur les mesures de similarité, le but étant de trouver pour chaque passage de la vidéo, le ou les paragraphes lui correspondant le mieux.

Le travail d'alignement se fait entre la transcription de la vidéo, découpée selon les slides, et le texte de l'article, découpé en paragraphe.

4.2.1 Comment définir la similarité de deux textes ?

Pour mesurer la similarité entre deux textes, nous allons comparer les mots qui y sont présents. En effet, lorsque deux textes parlent du même sujet, nous pouvons nous attendre à voir certains mots revenir.

Prenons l'exemple suivant. Nous avons la phrase "I am eating an apple", et le texte suivant : "I was sleeping under a tree. An apple fell from the tree. I ate the apple.". Laquelle des phrases du texte correspond le mieux à la phrase initiale ?

Dans cet exemple nous allons comparer la phrase (1) "I am eating an apple." avec les phrases (A) "I was sleeping under a tree.", (B) "An apple fell from the tree.", et (C) "I ate the apple.".

En pratique, ce ne sont pas des phrases qui seront comparées, mais les passages de la transcription, précédemment découpée, avec les paragraphes de l'article.

4.2.2 Un pré-traitement nécessaire

Le premier pré-traitement sert à découper les phrases en mots, en enlevant la ponctuation. C'est la *tokenisation*. Nous obtenons pour chaque phrase un tableau de mot :

(**1**) ["I", "am", "eating", "an", "apple"]

(**A**) ["I", "was", "sleeping", "under", "a", "tree"]

(**B**) ["An", "apple", "fell", "from", "the", "tree"]

(**C**) ["I", "ate", "the", "apple"]

Lorsque nous comparons des textes, il faut faire attention aux mots trop courants, comme les déterminants, ou les pronoms. Ces mots, appelés *stop words*, sont supprimés pour l'analyse.

(**1**) ["eating", "apple"]

(**A**) ["sleeping", "tree"]

(**B**) ["apple", "fell", "tree"]

(**C**) ["ate", "apple"]

Enfin, il nous paraît évident que "eating" et "ate" devraient être mis en relation, car ils ont la même racine. Pour cela, nous allons regrouper tous les mots d'une même famille dans sa forme réduite appelée *lemme*. Ainsi "eating" deviendra "eat" et "apples" deviendra "apple". Cette opération s'appelle la *lemmatisation*.

(1) [“eat”, “apple”]

(A) [“sleep”, “tree”]

(B) [“apple”, “fall”, “tree”]

(C) [“eat”, “apple”]

Ici, “fell” a été lemmatisé en “fall”. Cependant en pratique, la bibliothèque python utilisée (nltk) n’a pas réussi cette lemmatisation. Sur cet exemple précis, avoir “fell” ou “fall” n’a pas vraiment d’impact, mais en analysant les résultats, il faudra se rappeler que les traitements que nous effectuons peuvent être erronés.

Nous pouvons donc dresser la liste des mots utilisés dans l’ensemble des phrases analysées :

eat
apple
sleep
tree
fall

La phase de pré-traitement est terminée, nous pouvons maintenant commencer à analyser les mots restants.

4.2.3 Tf-idf

Le tf-idf donne une valeur à un mot dans un texte selon sa fréquence d’apparition dans le texte et sa fréquence d’apparition dans l’ensemble des textes analysés.

En effet, plus un mot apparaît fréquemment dans un texte, plus il a de l’importance pour ce texte. Mais si un mot apparaît fréquemment dans tous les textes analysés, alors il a moins d’importance pour un texte en particulier. Il faut prendre en compte ces deux aspects pour pondérer l’importance d’un mot pour un texte.

Le premier aspect, la fréquence d’apparition du mot dans le texte, est appelé tf (term frequency). Un tf est spécifique à un mot, dans un document, noté $tf_{t,d}$, avec t un terme et d un texte.

Nous allons donc, pour chaque texte analysé, compter le nombre d’apparition de chaque mot. Soit dans notre exemple :

— (1)

eat	1
apple	1
sleep	0
tree	0
fall	0

— (A)

eat	0
apple	0
sleep	1
tree	1
fall	0

— (B)

eat	0
apple	1
sleep	0
tree	1
fall	1

— (C)

eat	1
apple	1
sleep	0
tree	0
fall	0

Le deuxième aspect, la fréquence d'apparition du mot dans tous les textes analysés, est appelé df (document frequency). Ce df est spécifique à chaque mot, et est noté df_t, avec t un terme.

Il se calcule en comptant, pour chaque mot, son nombre d'apparition dans l'ensemble des textes analysés.

eat	2
apple	3
sleep	1
tree	2
fall	1

Plus un mot apparaît souvent dans l'ensemble des textes, c'est-à-dire plus il a un df élevé, moins il est significatif pour différencier ces textes. Cependant, si un mot apparaît significativement beaucoup de fois dans un texte, même s'il apparaît également souvent dans l'ensemble des textes, il peut être considéré comme important. Nous allons donc calculer l'idf (inverse document frequency).

$$idf_t = \log \frac{N}{df_t}$$

Le logarithme sert à atténuer l'importance du df par rapport au tf.

eat	0.301
apple	0.125
sleep	0.602
tree	0.301
fall	0.602

Le tf-idf est une pondération d'un mot dans un texte, qui prend en compte à la fois le tf et l'idf. Il s'applique donc à un terme t dans un texte d. Il se

calcule comme suit :

$$tfidf_{t,d} = tf_{t,d} * idf_t$$

Nous obtenons donc, pour chaque texte, un tableau de mots avec leur valeur de tf-idf associée.

— (1)

eat	0.301
apple	0.125
sleep	0.0
tree	0.0
fall	0.0

— (A)

eat	0
apple	0
sleep	0.602
tree	0.301
fall	0

— (B)

eat	0
apple	0.125
sleep	0
tree	0.301
fall	0.602

— (C)

eat	0.301
apple	0.125
sleep	0
tree	0
fall	0

Les formules présentées ici sont une façon usuelle de calculer le tf-idf. Il existe cependant plusieurs variantes, notamment pour le calcul du tf et de l'idf. Voici un tableau les recensant, trouvé dans le livre *"Introduction to Information Retrieval"* [1] page 118.

tf		idf	
naturel	$tf_{t,d}$	no	1
logarithm	$\begin{cases} 1 + \log tf_{t,d} & \text{si } tf_{t,d} > 0 \\ 0 & \text{sinon} \end{cases}$	idf	$\log \frac{N}{df_t}$
augmented	$0.5 + \frac{0.5 * tf_{t,d}}{\max(tf_{t,d})}$	prob idf	$\max\{0, \log \frac{N - df_t}{df_t}\}$
boolean	$\begin{cases} 1 & \text{si } tf_{t,d} > 0 \\ 0 & \text{sinon} \end{cases}$		
log ave	$\frac{1+\log(tf_{t,d})}{1+\log(ave_{t \in d}(tf_{t,d}))}$		

Nous utilisons actuellement la méthode *naturel* pour le tf, et la méthode *idf* pour l'idf. Mais d'autres méthodes, comme la méthode *logarithm* du tf pourrait être intéressante. Ceci sera discuté dans la partie résultat.

4.2.4 Mesure cosinus

Nous avons donc, pour chacun des textes analysés, un tableau de mots. Ce tableau peut alors être représenté mathématiquement comme un vecteur à n dimensions, n étant le nombre de mots.

Ces vecteurs peuvent être comparés deux à deux en regardant leurs angles. Ainsi, plus l'angle entre deux vecteurs est faible, plus les vecteurs sont proches

l'un de l'autre, et donc plus les textes correspondants sont similaires. Pour quantifier cette similarité, nous utilisons le cosinus. En effet, plus l'angle est faible, plus le cosinus de l'angle est fort ($\cos(0) = 1$). Un score de similarité compris entre 0 et 1 est alors obtenu.

(1) et (A) : 0.0

(1) et (B) : 0.07

(1) et (C) : 1.0

La phrase (1) "I am eating an apple" correspond donc parfaitement à la phrase (C) "I ate the apple". Elle correspond très peu à la phrase (B) "An apple fell from the tree" (les deux parlent de pomme), et pas du tout avec la phrase (A) "I was sleeping under a tree".

4.2.5 Un peu de contexte

Nous sommes donc capables de mesurer la similarité entre des passages de la vidéo et des paragraphes de l'article. Mais dans un discours, le contexte est généralement important, ce que nous n'avons pas du tout pris en compte. En effet, parfois un paragraphe peut nous intéresser si on prend en compte les paragraphes qui l'entourent, même si, tout seul, il n'a pas l'air très similaire au passage analysé.

Nous avons donc pensé à prendre en compte le contexte des passages et des paragraphes pour notre calcul de similarité. Pour cela, nous changeons le calcul du tf. Il ne vaut plus seulement le nombre d'apparition du mot dans le texte, mais ce nombre d'apparition plus le nombre d'apparition du mot dans les textes alentours, pondéré par un coefficient.

Par exemple, si, pour un terme t et un texte d, le nombre d'apparition d'un mot dans un texte est noté $na_{t,d}$, alors :

$$tf_{t,d} = na_{t,d} + 0.5na_{t,d-1} + 0.5na_{t,d+1} + 0.2na_{t,d-2} + 0.2na_{t,d+2}$$

Ainsi, certains paragraphes, qui nous intéressent pour l'alignement, pourront voir leur score de similarité augmenter grâce à leur contexte.

4.2.6 Réalisation de la mesure de similarité

Pour calculer les mesures de similarité, nous avons utilisé une bibliothèque python, scikit-learn [7], qui permet de calculer les tf-idf d'un ensemble de textes, puis de calculer des mesures cosinus.

7. `http://scikit-learn.org/stable/index.html`

Nous n'avons pas eu de problèmes pour la mesure cosinus, cependant les méthodes de calcul de tf-idf ne permettaient pas d'appliquer toutes les variantes désirées.

En effet, scikit-learn fournit trois objets, CountVectorizer, qui permet de compter les termes dans les textes, TfidfTransformer, qui à partir des fréquences des termes, calcule les tf-idf, et TfidfVectorizer, qui fait les deux à la fois.

Nous avons donc commencé par utiliser TfidfVectorizer, mais il a vite atteint ses limites. En effet, il ne permet pas de prendre en compte le contexte dans le calcul du tf-idf. Il a donc fallu le faire par nous-même.

Pour cela, nous avons utilisé CountVectorizer pour compter les mots, puis nous avons calculé nous-même les valeurs du tf, de l'idf, puis du tf-idf. Cette façon de faire permet une plus grande modularité, car nous contrôlons entièrement chaque partie du calcul.

Ainsi, pour calculer le tf d'un mot dans un texte, nous avons pu rajouter un tableau de pondération des textes alentours. De plus, si nous décidons d'utiliser une variante de calcul pour le tf ou l'idf, il est possible de l'implémenter en toute facilité.

Scikit-learn a une autre limitation. En effet il n'effectue aucune lemmatisation avant ses calculs. Pour la rajouter, il est possible de passer en argument un tokenizer qui sert à découper le texte en mots. Il sera utilisé par CountVectorizer à la place du tokenizer par défaut. Nous avons donc pu créer notre propre tokenizer, qui lemmatise en plus de tokeniser, en utilisant une deuxième bibliothèque python, nltk [8].

4.2.7 Les fichiers de sortie

Le script python fournit beaucoup de données en sortie. Il renvoie les mesures de similarité pour chaque couple passage de vidéo/paragraphe de l'article, bien sûr, mais aussi des informations contextuelles, pour évaluer les résultats. Ces informations sont stockées dans plusieurs fichiers xml.

Le fichier alignement.xml (cf annexe B.1 page 32) contient pour chaque passage de la vidéo, l'ensemble des paragraphes ayant une similarité non nulle, les mots en commun entre le paragraphe et le passage, ainsi que la moyenne, l'écart-type et le pourcentage de zéro des similarités pour ce passage.

Les fichiers infoSpeech.xml et infoParagraphe.xml (cf annexe B.2 page 33 et annexe B.3 page 34) indiquent respectivement pour chaque passage de la vidéo et chaque paragraphe, les mots qu'ils contiennent, ainsi que leurs

8. `www.nltk.org`

valeurs de tf et de tf-idf.

Le fichier vocabulary.xml (cf annexe B.4 page 35) contient tous les termes analysés, c'est-à-dire, les mots restants après les pré-traitements. Il indique, pour chaque mot, son df et son idf.

Enfin, le fichier infoLemmatizer.xml (cf annexe B.5 page 36) associe chaque mot trouvé à son lemme (issu de la lemmatisation).

4.3 Visualisation

Les mesures de similarité précédemment calculées sont des valeurs de comparaison entre un fragment de vidéo et un paragraphe. Avec cela, nous pouvons donc trouver le ou les paragraphes correspondant le mieux à un passage de la vidéo, et même ordonner les paragraphes d'après leur similarité à un passage donné.

4.3.1 Un problème d'évaluation

Le problème se pose quand il s'agit de savoir si les résultats des mesures de similarité donnent bien les informations désirées. Est-ce que le paragraphe désigné comme le plus similaire est bien celui qui est attendu pour l'alignement avec le passage de la vidéo ? Peut-être préférerions-nous le troisième plus similaire, ou le cinquième. Pire, peut-être que le paragraphe que nous souhaiterions aligner n'est même pas trouvé comme significativement similaire.

De plus, lorsque nous tentons d'améliorer nos algorithmes, est-il possible de valider une réelle amélioration des résultats ? Comment pouvons nous évaluer nos résultats ?

4.3.2 Le faire à la main

La principale façon de le faire proprement est d'écouter et de lire plusieurs fois la conférence et l'article, jusqu'à les maîtriser suffisamment pour connaître à l'avance les résultats attendus, ce qui est plutôt laborieux. De plus, cela ne pourra de toutes façons pas être fait pour la totalité des conférences existantes.

Pour aider à faire cette vérification, nous avons conçu une petite interface web permettant de visionner un passage de la vidéo, et proposant à côté les trois paragraphes ayant les meilleurs scores de similarité. Il faut ensuite cliquer sur le paragraphe qui nous paraît le plus approprié des trois. Ainsi, il est possible de valider (plus ou moins) les résultats rapidement. Cette

interface ne prend en revanche pas en compte le cas où le paragraphe désiré ne fait pas parti des trois proposés.

Nous avons déjà accès à un alignement qui a été fait à la main en regardant seulement les slides, pas la vidéo. Et nous pouvons aussi travailler sur une conférence donnée par Colin de la Higuera, qu'il va donc pouvoir annoter lui-même. Ainsi, il sera possible de comparer les résultats obtenus avec ceux attendus.

Cette technique reste de toutes façons très limitée, et ne permet pas d'évaluer tous les résultats possibles, sur la totalité des conférences existantes.

4.3.3 Trouver une mesure de comparaison

Il nous faut alors trouver une ou plusieurs mesures de comparaison. Nous avons donc travaillé sur une visualisation des résultats, en incluant un maximum d'informations pour déterminer celles qui pourraient nous intéresser.

L'interface permet de visualiser la timeline de la vidéo, découpée en fragments, ainsi que les paragraphes, page par page. Nous avons mis au point plusieurs modes d'affichage pour améliorer la visualisation. Nous pouvons ainsi tracer des traits entre les fragments et les paragraphes, ou bien jouer sur leur opacité au survol de la souris pour les faire ressortir.

Deux sliders permettent d'affiner l'affichage des informations. Il est possible de choisir le nombre maximum de paragraphes liés à afficher (les n meilleurs), et/ou de paramétrer un seuil pour la valeur de similarité (tous les liens qui ont une mesure de similarité supérieur à k).

Le texte d'un paragraphe ou d'un passage de la vidéo s'affiche au survol de la souris. De plus, cliquer sur un fragment de vidéo affiche les informations sur son alignement, et même les informations sur chaque mot.

Nous avons donc un moyen de visualiser rapidement et simplement tous les données obtenues lors du calcul des mesures de similarité. Cette visualisation nous sert à comprendre et évaluer les résultats de ces mesures.

5 Résultat

5.1 Découpage de la vidéo et de l'article

5.1.1 Découpage de l'article

En regardant juste la typographie de l'article, nous obtenons de bons résultats pour retrouver les paragraphes (voir annexe B.6 page 37).

Les paragraphes mal découpés sont ceux qui se trouvent sur deux pages, ou ceux qui sont coupés en deux par une formule mathématique ou une figure.

Nous trouvons en revanche énormément de bruit. Les titres sont reconnus comme des paragraphes, de même que les légendes, ou les formules mathématiques.

Les options de pré-traitement, comme la suppression des lignes qui n'ont pas la bonne marge ou la bonne police permet de supprimer la plupart de ces faux paragraphes. Nous avons pour l'instant décidé de les garder, notamment pour les titres, qui peuvent donner de bonnes informations.

Une idée à développer serait de garder ce bruit, mais de le différencier malgré tout des paragraphes eux-mêmes.

Pour améliorer encore le découpage, une approche typographique ne suffira plus. Il faudra commencer à analyser le texte, et regarder, par exemple, si la dernière ligne d'une page finit par un point et sinon, la raccrocher au début de la page suivante. Mais cela compliquera beaucoup le programme.

5.1.2 Découpage de la vidéo

Le découpage de la vidéo d'après le temps des slides pose plusieurs problèmes, mais présente quand même des avantages. Il est rapide et simple à mettre en place et, théoriquement, donne des morceaux cohérents.

Le principal problème est, qu'en pratique, le conférencier parle pendant qu'il change de slide. Il va donc tour à tour parler du slide suivant, avant de le montrer, ou parler du slide précédent après être passé au suivant.

Nous avons tenté de régler ce problème en trouvant le plus gros silence autour du changement de slide, mais les résultats ne sont pas toujours parfaits.

De plus, un conférencier peut parfois revenir sur un point précédent qu'il a oublié, ou parler d'un sujet qui n'a pas beaucoup de rapport avec le slide.

S'intéresser seulement aux slides permet donc d'avoir un découpage à peu près correct, de manière rapide, mais il serait intéressant de réfléchir à d'autres manières de découper la vidéo.

5.2 Les mesures de similarité

Les résultats des mesures de similarité sont extrêmement riches et durs à évaluer. C'est à cela que nous travaillons actuellement. Nous avons développé (et nous continuons à développer) une interface de visualisation des résultats, qui permet de voir rapidement les informations disponibles.

Pour l'instant, les résultats analysés ont été calculés sur une conférence dont la transcription a été faite par un humain, pour s'affranchir des erreurs de la transcription. Les mesures ont été effectuées sans prendre en compte le contexte.

Il y a déjà plusieurs remarques intéressantes à faire.

Premièrement, des titres reviennent assez souvent dans les paragraphes ayant les meilleures similarités, et ils ont souvent des scores significativement plus élevés que les autres paragraphes. Cela s'explique par l'importante concentration de mots-clés dans les titres, et par le fait que la longueur d'un texte influe sur la mesure de similarité, à cause de la normalisation induite par la mesure cosinus. Ainsi un paragraphe court, comme un titre, voit son score augmenter.

Dans notre contexte, où nous souhaitons trouver les paragraphes les plus similaires à un passage de la vidéo, nous préférerions sûrement obtenir des paragraphes riches en contenu. Cependant, les titres peuvent être intéressants, dans la mesure où ils peuvent être vus comme une référence au texte de leur section.

Dans cette mesure, plusieurs pistes peuvent être envisagées. Il serait déjà intéressant de voir les résultats que donnent les mesures de similarité en prenant en compte le contexte des paragraphes, et dans quelles mesures les titres influencent ces résultats.

Une deuxième piste serait de traiter les titres comme des entités à part. Ils ne seront alors plus considérés comme des paragraphes, et ne sortiront donc plus dans les résultats. Ils seront en revanche utilisés pour calculer les mesures de similarité, car ils fournissent des informations supplémentaires et pertinentes.

Un deuxième problème se pose à propos du poids de certains mots, qui influencent énormément le résultat. Par exemple, dans la conférence actuellement analysée, le mot "action" est présent très souvent. Il a donc un idf très faible, ce qui doit baisser son importance dans le calcul de la mesure de similarité. Mais lorsqu'il est présent dans un paragraphe, il y est souvent répété énormément de fois, ce qui augmente largement son poids. Ainsi, le mot "action" a souvent un poids très élevé pour déterminer les paragraphes

similaires, alors qu'il ne paraît pas si discriminant.

Pour régler ce problème, nous allons changer la formule de calcul du tf, pour essayer la formule logarithmique :

$$wf_{t,d} = \begin{cases} 1 + \log(tf_{t,d}) & \text{si } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{sinon} \end{cases}$$

Cette formule porte une grande importance sur la présence ou non d'un mot dans un segment. En effet, le nombre d'apparition du mot ne compte que de manière logarithmique, c'est-à-dire que le score sera de 0 si le mot n'est pas présent, de 1 s'il est présent une fois, de 3 pour 8 apparitions et il faudra 1024 apparitions du mot pour avoir un score de 10.

Cela permet de pondérer plus fortement la présence du mot, tout en regardant toujours un peu son nombre d'apparition.

Si nous regardons maintenant les cinq ou dix paragraphes les plus similaires à chaque passage de la vidéo, ils sont, pour la plupart, regroupés en un ou deux blocs. Cela dessine déjà un début d'alignement global, c'est donc un indice plutôt positif, même s'il reste à valider.

Les prochaines pistes d'améliorations porteront principalement sur une analyse plus approfondie des résultats. Nous essayerons de regarder les résultats obtenus sur des conférences dont nous possédons l'alignement, ce qui nous permettrait de les évaluer plus concrètement.

Il faut aussi voir ce que donne le calcul des mesures de similarité sur une transcription automatique, pour tester la robustesse de l'algorithme face aux erreurs de transcription.

Enfin, nous essayerons de faire nos calculs en prenant en compte le contexte des paragraphes. Nous pourrons alors étudier l'influence du contexte sur les résultats, et son utilité pour l'alignement.

6 Conclusion

En deux mois de stage, j'ai déjà appris énormément. J'ai eu un aperçu de deux domaines qui m'intéressent, la recherche et le traitement automatique du langage naturel. J'ai ainsi pu me familiariser avec des notions de TALN, comme les mesures de similarité et l'alignement, ce qui me sera utile dans ma poursuite d'étude en Master ATAL.

Ce stage n'est pas encore terminé, mais nous commençons à obtenir des résultats et à voir s'ouvrir de nombreuses pistes d'améliorations. Cela va nous permettre de nous intéresser au problème d'alignement global durant le dernier mois de mon stage.

Il y a des pistes qui ne seront pas abordées lors de ce stage par manque de temps, mais qui seraient intéressantes à essayer, comme l'apprentissage automatique. Cela offre encore de nombreuses perspectives dans le domaine de l'alignement multimodal de contenus éducatifs.

Bibliographie

Références

[1] Christopher D.Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Annexes

A Données

A.1 Transcription de la vidéo

```
<tt xml:lang="en" xmlns="http://www.w3.org/2006/04/ttaf1" xmlns
   :tts="http://www.w3.org/2006/10/ttaf1#style" xmlns:tl="
   translectures.eu">
<head>
<tl:d vI="/mnt/projects/translectures/videolectures/English/
   export_all/i/icgi08_higuera_llbr/video_01/
   x2inbxuydf53pt6c4pl3mi73g7g742yo.mp4.wav.xml1st#2013-06-21
   T19:58:28" aT="automatic" aI="emlDecoder" aC="0.70" wS="eml
   -length-based" tS="2013-06-21T19:58:28" cM="0.8200" b="0.00
   " e="1750.06" eT="12839.01" mI="
   transLectures_2_120810_160717" pS="SEG CN 1ST CLU SAT CN 2
   ND " aL="1761.86"/>
</head>
<body>
<tl:s sI="0" cM="0.7800" b="0.00" e="25.02">
<tl:w cM="1.0000" b="0.00" e="0.02">~SILENCE~</tl:w>
<tl:w cM="1.0000" b="0.14" e="0.60">you</tl:w>
<tl:w cM="0.5300" b="0.70" e="0.72">~SILENCE~</tl:w>
<tl:w cM="0.4200" b="0.82" e="1.10">know</tl:w>
<tl:w cM="0.5800" b="1.26" e="1.28">~SILENCE~</tl:w>
...
<tl:w cM="0.4100" b="24.01" e="24.29">some</tl:w>
<tl:w cM="0.5800" b="24.34" e="24.76">wanted</tl:w>
<tl:w cM="0.8300" b="24.88" e="25.02">half</tl:w>
</tl:s>
...
</body>
```

A.2 Transcription des slides

```
<tt xml:lang="en" xmlns="http://www.w3.org/2006/04/ttaf1" xmlns
   :tts="http://www.w3.org/2006/10/ttaf1#style" xmlns:tl="
   translectures.eu">
<head>
<tl:document aT="human" aI="UPV" aC="1.0" cM="1.0" b="0" e="
   1761.0"/>
</head>
<body>
<tl:s sI="1" b="0.0" e="53.279">
&lt;p&gt;&lt;/p&gt;
</tl:s>
<tl:s sI="2" b="53.279" e="68.221">
&lt;p&gt;the authors&lt;/p&gt;
                              &lt;p&gt;L&lt;/p&gt;
                              &lt;p&gt;'&lt;/p&gt;
                              &lt;p&gt;Jean Christophe&lt;/p&gt;
                              &lt;p&gt;janodet&lt;/p&gt;
                              &lt;p&gt;'&lt;/p&gt;
                              &lt;p&gt;2&lt;/p&gt;
</tl:s>
...
</body>
```

A.3 Données de l'article obtenues avec pdftohtml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pdf2xml SYSTEM "pdf2xml.dtd">

<pdf2xml producer="poppler" version="0.22.5">
<page number="1" position="absolute" top="0" left="0" height="
   1188" width="918">
    <fontspec id="0" size="24" family="Times" color="#000000"/>
    <fontspec id="1" size="15" family="Times" color="#000000"/>
     ...
    <fontspec id="8" size="13" family="Times" color="#000000"/>
    <image top="713" left="475" width="113" height="0" src="cmd
       -1_1.png"/>
    <image top="713" left="588" width="263" height="0" src="cmd
       -1_2.png"/>
    ...
    <image top="1061" left="475" width="143" height="0" src="
       cmd-1_39.png"/>
    <text top="101" left="149" width="616" height="37" font="0"
       ><b>Automatic Labeling of Multinomial Topic Models</b></
       text>
    <text top="172" left="275" width="365" height="24" font="1"
       >Qiaozhu Mei, Xuehua Shen, Chengxiang Zhai</text>
    ...
    <text top="43" left="81" width="158" height="21" font="8"><
       b>Research Track Paper</b></text>
</page>
...
</pdf2xml>
```

A.4 Découpage de l'article

```
<?xml version="1.0" encoding="UTF-8"?>
<pdf nbParagraphe="367" >
       <paragraphe id="0" begin="0.0" end="0.0">JMLR: Workshop
           and Conference Proceedings vol (2010) 1  21 24th
          Annual Conference on Learning Theory</paragraphe>
       <paragraphe id="1" begin="0.0" end="0.0">Minimax Regret
           of Finite Partial-Monitoring Games in Stochastic
          Environments </paragraphe>
       <paragraphe id="2" begin="0.0" end="0.0">G abor Bart ok
           bartok@cs.ualberta.ca D avid P al dpal@cs.ualberta.
          ca Csaba Szepesv ari szepesva@cs.ualberta.ca
          Department of Computing Science, University of
          Alberta, Edmonton, T6G 2E8, AB, Canada</paragraphe>
       <paragraphe id="3" begin="0.0" end="0.0">Editors: Sham
          Kakade, Ulrike von Luxburg</paragraphe>
       <paragraphe id="4" begin="0.0" end="0.0">Abstract</
          paragraphe>
       <paragraphe id="5" begin="0.0" end="0.0">In a partial
          monitoring game, the learner repeatedly chooses an
          action, the environment responds with an outcome,
          and then the learner suffers a loss and receives a
          feedback signal, both of which are fixed functions
          of the action and the outcome. The goal of the
          learner is to minimize his regret, which is the
          difference between his total cumulative loss and the
           total loss of the best fixed action in hindsight.
          Assuming that the outcomes are generated in an i.i.d
          . fashion from an arbitrary and unknown probability
          distribution, we characterize the minimax regret of
          any partial monitoring game with finitely many
          actions</paragraphe>
       ...
</pdf>
```

A.5 Découpage de la vidéo

```
<?xml version="1.0" ?>
<transcript duree="1378.71" nbSpeech="13">
        <speech begin="3.76" end="6.54" id="0" sI="1"> please I
           'll I'll start with a declining population because
           it's a little different than from from the previous<
           /speech>
        <speech begin="6.54" end="88.66" id="1" sI="2"> so
           consider Lerner and an environment of some repeated
           again staying in every time but the learner Tuesday'
           s action and the environment using an outcome that
           big their choices to Eritrea and the referred by the
            following things but referee populate feed that
           Anna based on the part of the action and the outcome
            and the feedback function and the bulk of its also
           based on the loss function and the the action and
           the outcome and it's important to know that these
           these functions are known to both the Lerner and
           environment and everybody and then In that in the
           next 2 days the referee gives the feedback to the
           Lerner and knowledgeable about the that the loss in
           that revealed the learner and in the thought we
           really care about finance the testifying at personal
            pursuing with finding many actions and outcomes and
            that was the case seeking comment meaning that the
           outcomes are chosen in an ID menu every time step </
           speech>
...
</transcript>
```

B Résultats

B.1 Alignement.xml

```
<?xml version="1.0" ?>
<alignment>
        <speech ecartType="0.000492773837929" id="0" moyenne="
           0.00126507761514" percentZero="97.8201634877">
                <paragraphe id="40" matchingWords="
                   471:3.24587911099" similarite="
                   0.0249851485397"/>
                <paragraphe id="39" matchingWords="
                   471:3.24587911099" similarite="
                   0.0385477210675"/>
                <paragraphe id="249" matchingWords="
                   1309:4.42118892403" similarite="
                   0.0437520215725"/>
                <paragraphe id="48" matchingWords="
                   1087:3.53745976891" similarite="
                   0.0449096488634"/>
                <paragraphe id="58" matchingWords="
                   1309:4.42118892403;1087:3.53745976891"
                   similarite="0.0565230310232"/>
                <paragraphe id="287" matchingWords="
                   471:3.24587911099" similarite="
                   0.0624469952179"/>
                <paragraphe id="121" matchingWords="
                   1087:3.53745976891" similarite="
                   0.0690811566284"/>
                <paragraphe id="49" matchingWords="
                   471:9.73763733298" similarite="0.124037761845
                   "/>
        </speech>
        <speech ecartType="0.00163686803805" id="1" moyenne="
           0.0153459334032" percentZero="57.2207084469">
                ...
        </speech>
        ...
</alignment>
```

B.2 InfoSpeech.xml

```
<?xml version="1.0" ?>
<infoSpeech>
        <speech id="0">
                <s_word idWord="856" tf="1.0" tf_base="1" tfidf=
                    "1.80163234623">little</s_word>
                <s_word idWord="1087" tf="1.0" tf_base="1" tfidf
                    ="1.88081359228">previous</s_word>
                <s_word idWord="1073" tf="1.0" tf_base="1" tfidf
                    ="2.57978359662">population</s_word>
                <s_word idWord="437" tf="1.0" tf_base="1" tfidf=
                    "2.57978359662">declining</s_word>
                <s_word idWord="471" tf="1.0" tf_base="1" tfidf=
                    "1.80163234623">different</s_word>
                <s_word idWord="4" tf="1.0" tf_base="1" tfidf="
                    1.57978359662">'s</s_word>
                <s_word idWord="1309" tf="1.0" tf_base="1" tfidf
                    ="2.1026623419">start</s_word>
                <s_word idWord="1" tf="2.0" tf_base="2" tfidf="
                    5.15956719323">'ll</s_word>
        </speech>
        <speech id="1">
                ...
        </speech>
        ...
</infoSpeech>
```

B.3 InfoParagraphe.xml

```
<?xml version="1.0" ?>
<infoParagraphe>
        <paragraphe id="0">
                <p_word idWord="67" tf="1.0" tf_base="1" tfidf="
                   2.57978359662">24th</p_word>
                <p_word idWord="782" tf="1.0" tf_base="1" tfidf=
                   "2.57978359662">jmlr</p_word>
                <p_word idWord="1538" tf="1.0" tf_base="1" tfidf
                   ="2.27875360095"></p_word>
                <p_word idWord="186" tf="1.0" tf_base="1" tfidf=
                   "1.80163234623">annual</p_word>
                <p_word idWord="378" tf="2.0" tf_base="2" tfidf=
                   "3.46937111321">conference</p_word>
                <p_word idWord="1461" tf="1.0" tf_base="1" tfidf
                   ="2.57978359662">vol</p_word>
                <p_word idWord="1098" tf="1.0" tf_base="1" tfidf
                   ="1.67669360962">proceeding</p_word>
                <p_word idWord="26" tf="1.0" tf_base="1" tfidf="
                   0.593011862351">1</p_word>
                <p_word idWord="1385" tf="1.0" tf_base="1" tfidf
                   ="1.7346855566">theory</p_word>
                <p_word idWord="834" tf="1.0" tf_base="1" tfidf=
                   "1.53839091146">learning</p_word>
                <p_word idWord="1481" tf="1.0" tf_base="1" tfidf
                   ="2.57978359662">workshop</p_word>
                <p_word idWord="7" tf="1.0" tf_base="1" tfidf="
                   0.30794199008">)</p_word>
        </paragraphe>
        <paragraphe id="1">
                ...
        </paragraphe>
        ...
</infoParagraphe>
```

B.4 Vocabulary.xml

```
<vocabulary>
        <word df="5" id="384" idf="1.88081359228">consider</
           word>
        <word df="1" id="314" idf="2.57978359662">chinese</word
           >
        <word df="1" id="168" idf="2.57978359662">ali</word>
        <word df="7" id="497" idf="1.7346855566">dynamic</word>
        <word df="1" id="1491" idf="2.57978359662">yellow</word
           >
        <word df="1" id="945" idf="2.57978359662">month</word>
        <word df="1" id="1552" idf="2.57978359662">path</word>
        <word df="2" id="804" idf="2.27875360095">kl-divergence
           </word>
        <word df="1" id="27" idf="2.57978359662">1+x</word>
        <word df="1" id="639" idf="2.57978359662">generalize</
           word>
        <word df="1" id="1544" idf="2.57978359662">shifted</
           word>
        <word df="1" id="613" idf="2.57978359662">follow</word>
        <word df="1" id="175" idf="2.57978359662">alt</word>
        <word df="1" id="1182" idf="2.57978359662">research</
           word>
        <word df="1" id="281" idf="2.57978359662">calculate</
           word>
        <word df="1" id="1411" idf="2.57978359662">tv</word>
        <word df="1" id="1107" idf="2.57978359662">program</
           word>
        <word df="6" id="1380" idf="1.80163234623">th</word>
        ...
</vocabulary>
```

B.5 InfoLemmatizer.xml

```
<?xml version="1.0" ?>
<infoLemmatizer>
        <l_word original="dynamic">dynamic</l_word>
        <l_word original="yellow">yellow</l_word>
        <l_word original="four">four</l_word>
        <l_word original="kl-divergence">kl-divergence</l_word>
        <l_word original="whose">whose</l_word>
        <l_word original="calculate">calculate</l_word>
        <l_word original="under">under</l_word>
        <l_word original="pride">pride</l_word>
        <l_word original="regrets">regret</l_word>
        <l_word original="@">@</l_word>
        <l_word original="conjecture">conjecture</l_word>
        <l_word original="incurs">incurs</l_word>
        <l_word original="1,2">1,2</l_word>
        <l_word original="every">every</l_word>
        <l_word original="jacob">jacob</l_word>
        <l_word original="solver">solver</l_word>
        <l_word original="path">path</l_word>
        <l_word original="solution">solution</l_word>
        <l_word original="convenience">convenience</l_word>
        <l_word original="vector">vector</l_word>
        ...
</infoLemmatizer>
```

B.6 Découpage de l'article

where in (4) we used that $O_k(n)$ is a unit vector and $\mathrm{E}_{n-1}[O_k(n)]$ is a probability vector.

For i, j non-neighboring cells, let $i = i_0, i_1, \ldots, i_r = j$ the path used for the estimate in round n. Then $\mu_{(i,j)}(n)$ can be written as

$$\mu_{(i,j)}(n) = \sum_{s=1}^{r} \mu_{(i_{s-1},i_s)}(n) = \sum_{s=1}^{r} \sum_{k \in A_{i_{s-1},i_s}} O_k(n)^\top v_{(i_{s-1},i_s),k} .$$

It is not hard to see that an action can only be in at most two neighborhood action sets in the path and so the double sum can be rearranged as

$$\sum_{k \in \bigcup A_{i_{s-1},i_s}} O_k(n)^\top (v_{(i_{s_k-1},i_{s_k}),k} + v_{(i_{s_k} i_{s_k+1}),k}) ,$$

and thus $\mathbb{V}\mathrm{ar}_{n-1}\left(\mu_{(i,j)}(n)\right) \le 2\sum_{s=1}^{r} \|v_{(i_{s-1},i_s)}\|_2^2 \le 2\sum_{\{i,j \text{ neighbors}\}} |v_{(i,j)}|_2^2$. ■

Lemma 7 *The range of the estimates $\mu_{(i,j)}(n)$ is upper bounded by $R = \sum_{\{i,j \text{ neighbors}\}} \|v_{(i,j)}\|_1$.*

Proof The bound trivially follows from the definition of the estimates. ■

Let δ be the confidence parameter used in BStopStep. Since, according to Lemmas 5, 6 and 7, $(\mu_{(i,j)})$ is a "shifted" martingale difference sequence with conditional mean $\alpha_{i,j}$, bounded conditional variance and range, we can apply Lemma 11 stated in the Appendix. By the union bound, the probability that any of the confidence bounds fails during the game is at most $N^2\delta$. Thus, with probability at least $1 - N^2\delta$, if BStopStep returns true for a pair (i, j) then $\mathrm{sgn}(\alpha_{i,j}) = \mathrm{sgn}(\mu_{(i,j)})$ and the algorithm eliminates all the actions whose cell is contained in the closed half space defined by $\mathcal{H} = \{p : \mathrm{sgn}(\alpha_{i,j}) p^\top (\ell_i - \ell_j) \le 0\}$. By definition $\alpha_{i,j} = (\ell_i - \ell_j)^\top p^*$. Thus $p^* \notin \mathcal{H}$ and none of the eliminated actions can be optimal under p^*.

From Lemma 11 we also see that, with probability at least $1 - N^2\delta$, the number of times τ_i^* the algorithm experiments with a suboptimal action i during the elimination phase is bounded by

$$\tau_i^* \le \frac{c(\mathbf{G})}{\alpha_{i,j^*}^2} \log \frac{R}{\delta \alpha_{i,j^*}} = T_i , \tag{5}$$

where $c(\mathbf{G}) = C(V + R)$ is a problem dependent constant.

The following lemma, the proof of which can be found in the Appendix, shows that degenerate actions will be eliminated in time.

Lemma 8 *Let action i be a degenerate action. Let $A_i = \{j : C_j \in \mathcal{C}, C_i \subset C_j\}$. The following two statements hold:*

1. *If any of the actions in A_i is eliminated, then action i is eliminated as well.*
2. *There exists an action $k_i \in A_i$ such that $\alpha_{k_i,j^*} \ge \alpha_{i,j^*}$.*