

# Alignement multimodal de contenus éducatifs

Un stage de recherche en TALN

Matthieu Riou

10 juin 2014

Université de Nantes  
Licence 3 Informatique

## Présentation du Stage

- Stage de L3 Informatique
- Stage en recherche
- Du 14 Avril au 11 Juillet
- Au LINA, dans l'équipe TALN

## Contexte du Stage

- Maîtres de stage : Colin de la Higuera, Solen Quiniou, Olivier Aubert
- Encadrante universitaire : Irena Rusu

## La plateforme COCo

- COCo pour CominOpenCourseware
- Mise à disposition de contenus pédagogiques
- Des contenus dans plusieurs formats
- Tournant autour de la vidéo : la vidéo augmentée

## La plateforme COCo

- COCo pour CominOpenCourseware
- Mise à disposition de contenus pédagogiques
- Des contenus dans plusieurs formats
- Tournant autour de la vidéo : la vidéo augmentée

## Les problématiques

- Beaucoup de contenus, difficiles à trier
- Des contenus riches, sous différentes formes
- Améliorer l'expérience de l'apprentissage en ligne

## Naviguer facilement entre différentes ressources éducatives

- Créer des liens entre les différentes ressources
  - Repérer les segments qui abordent un même sujet, une même idée
  - Créer des interfaces utilisateurs permettant de naviguer entre ces segments

## Naviguer facilement entre différentes ressources éducatives

- Créer des liens entre les différentes ressources
  - Repérer les segments qui abordent un même sujet, une même idée
  - Créer des interfaces utilisateurs permettant de naviguer entre ces segments

## L'alignement multimodal

- Faire correspondre des segments de deux documents
- Multimodal = Entre plusieurs formats
  - Dans ce stage, entre une vidéo et un texte
  - Utilisation de la transcription de la vidéo
  - Prise en compte des erreurs

## Trouver un alignement global

- Comparaison de documents similaires
  - Même sujet
  - Plan similaire
  - Changements dans l'ordre des parties
  - Les petites différences ont moins d'importance
- Le but final : faire correspondre les différentes parties des deux documents dans leur ensemble

# L'alignement

## Trouver un alignement global

- Comparaison de documents similaires
  - Même sujet
  - Plan similaire
  - Changements dans l'ordre des parties
  - Les petites différences ont moins d'importance
- Le but final : faire correspondre les différentes parties des deux documents dans leur ensemble

## Trois étapes

- Découper la vidéo et le texte en segments cohérents
- Comparer les segments de la vidéo et du texte
- Trouver une correspondance global entre la vidéo et le texte



# Découper la vidéo en segments cohérents

## Comment ?

- Découpage d'après le temps des slides
  - Chaque slide délimite une idée, une partie du discours
- Problème : les limites sont floues
- Découpage placé sur le plus grand silence autour du changement de slide

# Découper la vidéo en segments cohérents

## Comment ?

- Découpage d'après le temps des slides
  - Chaque slide délimite une idée, une partie du discours
- Problème : les limites sont floues
- Découpage placé sur le plus grand silence autour du changement de slide

## Résultats

- Résultats imparfaits, approximatifs
- Avantages : Simple et rapide

# Découper le texte en segments cohérents

## Comment ?

- Découpage en paragraphes
  - Unité de texte standard
  - Suffisamment grand pour contenir assez d'informations
  - Suffisamment petit pour se concentrer une idée précise
- Utilisation seulement de la typographie
  - Alinéa
  - Écart entre deux paragraphes

# Découper le texte en segments cohérents

## Comment ?

- Découpage en paragraphes
  - Unité de texte standard
  - Suffisamment grand pour contenir assez d'informations
  - Suffisamment petit pour se concentrer une idée précise
- Utilisation seulement de la typographie
  - Alinéa
  - Écart entre deux paragraphes

## Résultats

- Résultats satisfaisants
- Paragraphes bien découpés
- Parfois trop découpés

# Découper le texte en segments cohérents

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

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

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

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

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

and thus  $\text{Var}_{n-1}(\mu_{(i,j)}(n)) \leq 2 \sum_{s=1}^r \|v_{(i_{s-1}, i_s)}\|_2^2 \leq 2 \sum_{\{i,j \text{ neighbors}\}} \|v_{(i,j)}\|_2^2$ . ■

**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  $\delta$  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  $\alpha_{i,j}$ ,

# Comparer les segments

## Mesure de similarité

- Quantifier la similarité entre deux segments
  - Donner une valeur pour chaque couple de segments
  - Plus la valeur est forte, plus ils sont similaires
  - Permet d'ordonner les paragraphes d'après leur similarité à un passage de la vidéo
- Utilisée en fouille d'information

## La similarité

- Similaires = Parlent de la même idée
- Comparaison des mots présents dans les segments

## Un exemple

- ① “I am eating an apple”
- Ⓐ “I was sleeping under a tree”
- Ⓑ “An apple fell from the tree”
- Ⓒ “I ate the apple”

La phrase ① correspond-elle plus à la phrase Ⓐ, Ⓑ ou Ⓒ?

# Comparer les segments

## Pré-traitement

- Tokenisation : Découpage en mots
- Suppression des mots trop courants, les stop words
- Lemmatisation : Chaque mot est remplacé par sa racine



## Un exemple - Tokenisation

- ① [ "I", "am", "eating", "an", "apple" ]
- Ⓐ [ "I", "was", "sleeping", "under", "a", "tree" ]
- Ⓑ [ "An", "apple", "fell", "from", "the", "tree" ]
- Ⓒ [ "I", "ate", "the", "apple" ]

## Un exemple - Stop words

- ① [ “eating”, “apple” ]
- Ⓐ [ “sleeping”, “tree” ]
- Ⓑ [ “apple”, “fell”, “tree” ]
- Ⓒ [ “ate”, “apple” ]

## Un exemple - Lemmatisation

- ① [ “eat”, “apple” ]
- Ⓐ [ “sleep”, “tree” ]
- Ⓑ [ “apple”, “fall”, “tree” ]
- Ⓒ [ “eat”, “apple” ]

## Tf-idf

- Donner une valeur au mot
  - Selon sa fréquence d'apparition dans le segment étudié
  - Selon sa fréquence d'apparition dans l'ensemble des segments
- Plus il apparaît dans un segment donné, plus il est significatif pour ce segment
- Plus il apparaît dans l'ensemble des segments, moins il est significatif pour un segment donné

## Un exemple - Tf-idf

①	eat	0.301
	apple	0.125
	sleep	0.0
	tree	0.0
	fall	0.0
②	eat	0
	apple	0.125
	sleep	0
	tree	0.301
	fall	0.602

③	eat	0
	apple	0
	sleep	0.602
	tree	0.301
	fall	0
④	eat	0.301
	apple	0.125
	sleep	0
	tree	0
	fall	0

## Mesure cosinus

- Vecteurs de mots à  $n$  dimensions ( $n$  mots)
- Possible de regarder l'angle entre deux vecteurs
  - Plus l'angle est petit, plus les vecteurs sont proches
  - Donc plus les segments sont similaires
- On utilise le cosinus de l'angle
  - Valeur croissante
  - Entre 0 et 1

## Un exemple - Mesure cosinus

① et ① : 0.0

① et ② : 0.07

① et ③ : 1.0

## Résultats

- Une valeur de similarité pour chaque couple de segments vidéo/texte
- Un ordre des paragraphes les plus similaires pour un passage de vidéo donné
- Très dur à évaluer



## Résultats

- Une valeur de similarité pour chaque couple de segments vidéo/texte
- Un ordre des paragraphes les plus similaires pour un passage de vidéo donné
- Très dur à évaluer

## Évaluation

- À la main
- En trouvant des indicateurs
  - Moyenne, écart-type, pourcentage de zéro, ...
- Besoin d'interfaces de visualisation

# Mesure de similarité

## Résultats

- Une valeur de similarité pour chaque couple de segments vidéo/texte
- Un ordre des paragraphes les plus similaires pour un passage de vidéo donné
- Très dur à évaluer

## Évaluation

- À la main
- En trouvant des indicateurs
  - Moyenne, écart-type, pourcentage de zéro, ...
- Besoin d'interfaces de visualisation

## Visualisation

- Rapide et complète
- Quelques bonnes nouvelles

# Travail à venir

## Analyse des résultats

Trouver un critère d'évaluation satisfaisant

Améliorer la visualisation

## Améliorations

- Découpage des segments
- Mesure de similarité

## Et après?

- L'alignement global
- Des interfaces

# Conclusion

## Bilan intermédiaire

- Des résultats encourageants
- Encore du travail à faire

## Un stage enrichissant

- Notions de TALN
- Domaine de la recherche

## Tf-idf, un calcul en deux étapes

- Tf = Term frequency, le nombre d'apparition du mot dans le segment étudié
- Idf = Inverse Document Frequency, l'inverse du nombre d'apparition du mot dans le segment étudié
- $Tf - idf = Tf * Idf$

# Mesure de similarité

	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	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$	

## Des bases

- Commencé par Colin de la Higuera
- But : aligner les plus gros groupe de segments possibles
- Création d'un vecteur à 4 dimensions  $T[i, j, k, l]$ 
  - $[i, j]$  Les segments  $i$  à  $j$  du premier document
  - $[k, l]$  Les segments  $k$  à  $l$  du deuxième document
- Calcul des coûts d'alignements pour chaque couple de groupe possible.
- Sélection des alignements les moins coûteux avec les plus grands groupes possibles

# Alignement global

## Des bases

- Commencé par Colin de la Higuera
- But : aligner les plus gros groupe de segments possibles
- Création d'un vecteur à 4 dimensions  $T[i, j, k, l]$ 
  - $[i, j]$  Les segments  $i$  à  $j$  du premier document
  - $[k, l]$  Les segments  $k$  à  $l$  du deuxième document
- Calcul des coûts d'alignements pour chaque couple de groupe possible.
- Sélection des alignements les moins coûteux avec les plus grands groupes possibles

## Résultats

- Un algorithme implémenté
- Très peu efficace :  $O(n^4)$