Similarité entre les mots

Traitement Automatique des Langues
Master Informatique
Université Paris-Est Marne-la-Vallée

Matthieu Constant

Références de base du cours

Christopher D. Manning and Hinrich Schütze, 1999, *Foundations of Statistical Natural Language Processing*, Massachussetts Insititute of Technology

Ruslan Mitkov, 2003, *The Oxford Handbook of Computational Linguistics*, Oxford University Press

Plan

- introduction
- mesures de similarité
- construction d'une matrice similarité
- applications

Plan

- introduction
- mesures de similarité
- construction d'une matrice similarité
- applications

INTRODUCTION

Similarité entre les mots

- un objectif du TAL : représenter le sens d'un texte
  - besoin d'une analyse profonde des phrases
  - difficile !
- une méthode plus superficielle : se concentrer sur les mots
  - calculer leur degré de similarité
  - simple !

Applications

- expansion d'une requête pour la recherche d'information

- lever l'ambiguité de sens des mots
- assigner aux mots des catégories thématiques

- trouver les thèmes abordés dans un texte
- extraire les mots les plus importants (cf. projet)

- généralisation de comportements syntaxiques

Modèle

- espaces vectoriels : modèles simples et efficaces
  cf. cours précédents

- un mot peut être représenté dans un espace multi-dimensionnel

- similarité = similarité entre deux vecteurs
  (ex. cosinus)

Exemple 1 : représentation dans un espace de documents

	cosmonaut	astronaut	moon	car	truck
d1	1	0	1	1	0
d2	0	1	1	0	0
d3	1	0	0	0	0
d4	0	0	0	1	1
d5	0	0	0	1	0
d6	0	0	0	0	1

A: matrice documents-mots

Exemple 2 : représentation dans un espace de mots

- un mot peut être représenté dans un espace de mots
- Soit la matrice B définie telle que:

  B_{ij} est le nombre de fois que les mots *i* et *j* apparaissent ensemble dans un document

- B est une matrice de cooccurrence
- Remarque : $B = A^T.A$

Exemple 2

	cosmonaut	astronaut	moon	car	truck
cosmonaut	2	0	1	1	0
astronaut	0	1	1	0	0
moon	1	1	2	1	0
car	1	0	1	3	1
truck	0	0	0	1	2

B : matrice mots-mots

Exemple 3 : représentation dans un espace de modifieurs

- un nom peut être représenté dans un espace de modifieurs (ex. adjectifs)

- soit la matrice C définie telle que:

  C_{ij} est le nombre de fois que le nom *j* est modifié par le mot *i*

- exemple

  John has (GN [a.DET] [red.A] [car.N] GN).

Exemple 3

	cosmonaut	astronaut	moon	car	truck
Soviet	1	0	0	1	1
American	0	1	0	1	1
spacewalking	1	1	0	0	0
red	0	0	0	1	1
full	0	0	1	0	0
old	0	0	0	1	1

C: matrice modifieurs-noms

Remarques

- différents espaces de représentations donne des types de similarités différentes :
  - représentation par documents ou mots donne des similarités thématiques
    - ex. {astronaut, cosmonaut, moon} => SPACE EXPLORATION
  - représentation par modifieurs est plus précis
    - le modifieur donne des propriétés du nom
    - ex. *cosmonaut* et *astronaut* ont les mêmes propriétés mais pas *cosmonaut* et *moon*.

Remarques - 2

- les représentations par documents ou mots nécessitent peu d'outils linguistiques

  au pire, un étiqueteur morphosyntaxique avec lemmatiseur

- la représentation par modifieurs nécessite un analyseur syntaxique

  au mieux, analyseur syntaxique de surface

Plan

- introduction
- mesures de similarité
- construction d'une matrice similarité
- applications

MESURES DE SIMILARITE

Représentation par vecteurs binaires

- représentation la plus simple des mots : les composantes sont nulles ou non-nulles

- un vecteur binaire peut être représenté par l'ensemble des composantes du vecteur qui ont une valeur non nulle

- exemple :

  cosmonaut = {cosmonaut, moon, car}

Représentation par vecteurs binaires - 2

- calcul des similarités par des opérations sur les ensembles

- notation :

  Soit X un ensemble de mots

  card(X) est le nombre de mots de l'ensemble X

Mesures de similarité pour vecteurs binaires - 1

- coefficient de Dice (Dice coefficient)

$$dice(X,Y)=\frac{2.\text{card}(X\cap Y)}{card(X)+card(Y)}$$

- coefficient de Jaccard (Jaccard coefficient)

$$jaccard(X,Y)=\frac{card(X\cap Y)}{card(X\cup Y)}$$

Mesures de similarité pour vecteurs binaires - 2

- dice(X,Y) et jaccard(X,Y) se ressemblent et sont des réels compris entre 0.0 et 1.0
- bornes :
  - cas **0.0** : recouvrement nul
  - cas **1.0** : recouvrement parfait
- la mesure de Jaccard a tendance à plus pénaliser les cas avec peu de mots en commun, par rapport à la mesure de Dice

Mesures de similarité pour vecteurs binaires - 3

- coefficient de recouvrement (overlap coefficient)

$$overlap(X,Y)=\frac{card(X\cap Y)}{min(card(X),card(Y))}$$

- ce coefficient mesure en quelque sorte l'inclusion d'un ensemble dans un autre :

  le coefficient est 1.0 si l'un des deux ensembles est inclus dans l'autre

Mesures de similarité pour vecteurs binaires - 4

- cosinus

$$cosinus(X,Y)=\frac{card(X\cap Y)}{\sqrt{card(X).card(Y)}}$$

- *cosinus* et *dice* sont identiques si card(X) = card(Y)

- *cosinus* pénalise moins pour les ensembles de tailles très différentes
  => propriété intéressante

Vecteurs à valeurs réelles

- meilleur représentation des objets linguistiques par vecteurs à valeurs réelles

- dans un espace de dimension *n*,
  - un vecteur *x* est representé par *n* composantes réelles ;
  - x_i est la i^e composante de *x*

- longueur d'un vecteur *x* dans un espace euclidien :

$$|x| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

similarité de vecteurs à valeurs réelles

- on utilise la mesure du cosinus
- soient *x* et *y* deux vecteurs

$$\cos(x, y) = \frac{x.y}{|x|.|y|}$$

$$avec \quad x.y = \sum_{i=1}^{n} x_i . y_i$$

Plan

- introduction
- mesures de similarité
- construction d'une matrice similarité
- applications

CONSTRUCTION D'UNE MATRICE DE SIMILARITE

Prétraitement

- pour la suite, on suppose que l'on a fait, sur les documents traités, les prétraitements suivants :
  - tokenisation
  - étiquetage morphosyntaxique et lemmatisation
  - filtrage suivant catégorie grammaticale
    - ex. on garde les verbes, les noms et les adjectifs
    - ex. on supprime les mots grammaticaux

Prétraitement - 2

- texte :
  Le juge a condamné les accusés à cinq ans de prison.

- prétraitements:
  - tokenisation
  - étiquetage morphosyntaxique et lemmatisation
  - filtrage des mots grammaticaux (ex. prépositions, déterminants, les auxiliaires *avoir* et *être*, ...)

Prétraitement - 3

- Etiquetage morphosyntaxique et lemmatisation

  [*le*.DET] [*juge*.N] [*avoir*.V] [*condamner*.V] [*le*.DET] [*accusé*.N] [*à*.PREP] [*cinq*.DET] [*an*.N] [*de*.PREP] [*prison*.N]

- filtrage des mots grammaticaux

  [*juge*.N] [*condamner*.V][*accusé*.N][*an*.N][*prison*.N]

- texte prétraité

  [*juge*, *condamner*, *accusé*, *an*, *prison*]

Similarité et cooccurrence

- il existe un lien sémantique entre deux mots cooccurrents
  => la cooccurrence est utile pour le calcul de la similarité

- deux types de cooccurrences :
  - cooccurrence de premier ordre
  - cooccurence de deuxième ordre

Cooccurrence de premier ordre

- définition :
  - deux mots sont cooccurrents (de premier ordre) s'ils apparaissent ensemble dans une même fenêtre
  - une fenêtre peut être :
    - de longueur fixe (un nombre fixe de mots)
    - de longueur variable
      - phrase
      - paragraphe
      - document
  - un texte peut être vu comme une séquence de fenêtres

Exemple

- fenêtre : la phrase

- texte :

  Le juge a condamné Max à 10 ans de prison. Son avocat a fait appel du jugement.

- cooccurents de premier ordre après étiquetage morphosyntaxique, lemmatisation et filtrage :

  cooc(juge) = {condamner, Max, an, prison}
  cooc(avocat) = {appel, jugement}

Cooccurrence de deuxième ordre

- principe :

  deux mots ayant des contextes de cooccurrence (de premier ordre) proches (c.a.d. partageant un grand nombre de cooccurrents) sont proche sémantiquement.

- autre terme : cooccurrence indirecte

- la similarité entre deux mots peut donc donc définie par la cooccurrence indirecte entre deux mots.

Exemple

- exemple :
  L' institutrice donne une leçon de Mathématiques.
  La maîtresse donne une leçon de Mathématiques.

- les mots *institutrice* et *maîtresse* sont similaires car ils sont utilisés dans le même contexte :

  cooc(institutrice) = {lecon,Mathématiques}
  cooc(maîtresse) = {lecon,Mathématiques}

  cooc(institutrice) = cooc(maîtresse)

Poids de cooccurrence de premier ordre

- trouver une mesure adéquate pour donner un poids aux cooccurrents (de premier ordre) d'un mot
- en général, ces mesures sont basées sur la fréquence de cooccurrence dans un gros corpus d'apprentissage
- Soit $f(x,y)$ la fréquence de cooccurrence des mots x et y
  - $f(x,y)$ est le nombre de fois que le mots x et y apparaissent ensemble dans le corpus
  - deux mots apparaissent ensemble s'ils se trouvent dans une même fenêtre

Poids de cooccurrence de premier ordre - 2

- Soit $p(x,y)$ le poids de cooccurrence entre x et y

- différents calculs possibles :
  - fréquence de cooccurrence : $p(x,y) = f(x,y)$
  - fréquence de cooccurrence corrigée : $1 + \log f(x,y)$
  - mesure de Dice :

$$p(x,y) = \frac{2.\mathrm{f}(x,y)}{f(x)+f(y)}$$

$f(u)$ est la fréquence du mot u dans le corpus

Représentation d'un mot

- un mot *x* est un vecteur dont les composantes réelles correspondent à un mot appartenant au corpus
- la composante correspondant au mot *y* a pour poids $p(x,y)$
- en pratique,
  - on ne garde que les *n* meilleurs poids (ex. $n = 20$),
  - les autres sont mis à 0.

Exemple : emissions - 1

- corpus d'apprentissage
  - apprentissage sur un an du *New York Times*
  - prétraitement :
    - mise en minuscule
    - on ne garde que les noms
    - pas de lemmatisation
    - repérage de noms composés et entités nommées
- fenêtre : la phrase
- exemple : *emissions*

Exemple : emissions - 2

- les cooccurrents les plus fréquents de *emissions*

emissions,442
mr,55
percent,52
carbon dioxide,50
global warming,40
greenhouse gas,35
companies,29
year,28
carbon emissions,27
administration,25
bush,24
...

Exemple : emissions - 3

- meilleurs coocurrents de *emissions* (Dice) :

  emissions,0.99
  carbon dioxide,0.12
  global warming,0.09
  greenhouse gas,0.08
  carbon emissions,0.07
  greenhouse gas,0.06
  pollutants,0.05
  power plants,0.05
  gases,0.05
  pollution,0.04
  ...

Similarité

- la similarité entre deux mots *x* et *y* est une mesure de similarité de leur vecteurs :
  - le plus courant, cosinus
  - coefficient de Dice (si on considère *x* et *y* comme des vecteurs binaires)
  - ...

- on parle de **similarité distributionnelle** !

Matrice de similarité

- la matrice de similarité M est définie comme suit : M_{ij} est la similarité entre le mot *i* et le mot *j*

- M peut aussi être vue comme un graphe :
  - les sommets sont des mots
  - les arcs entre les mots correspondent à leurs similarités

Exemple - 1

- exemple tiré de (Manning et Schütze, 1999)

- corpus d'apprentissage :
  - 4 mois du NYT (environ 14 millions de mots)
  - prétraitement : tokenisation uniquement

- procédure :
  - les 20,000 mots (resp. 10,000) les plus fréquents en ligne (resp. colonne) après élimination des 100 plus fréquents
  - poids des composantes des vecteurs : $1 + \log f(x,y)$
  - fenêtre : 50 mots
  - mesure de similarité : cosinus

Exemple - 2

mot	**mots les plus similaires**							
garlic	sauce	.732	pepper	.728	salt	.726	cup	.726
fallen	fell	.932	decline	.931	rise	.930	drop	.929
engineered	genetically	.758	drugs	.688	research	.687	drug	.685
Alfred	named	.814	Robert	.809	William	.808	W	.808
simple	something	.964	things	.963	You	.963	always	.962

- marche bien pour *garlic* et *fallen* !
- dépendence au corpus pour *engineered*
- marche très mal pour mots distribués tout au long du corpus (*Alfred*, *simple*)

Plan

- introduction
- mesures de similarité
- construction d'une matrice similarité
- applications

APPLICATIONS

Expansion de requêtes

- préciser une requête :

  proposer à l'utilisateur d'**étendre sa requête** en sélectionnant des mots choisis dans la **liste des meilleurs cooccurrents** du premier ordre des mots de la requête

- exemple :
  - requête = emissions
  - choix de mots pour expansion

    carbon dioxide, global warming, greenhouse gas, carbon emissions, greenhouse gas, pollutants, power plants

Expansion de requêtes - 2

- étendre la portée d'une requête : expansion par des mots similaires (quasi-synonymes)
- réduction du silence, alternative fiable aux dictionnaires de synonymes
- exemple :
  - requête : astronaut
  - suggestion de ajout de *cosmonaut* car ses deux mots sont proches

Généralisation de comportements syntaxiques

- hypothèse : deux mots sémantiquement proches ont tendance à avoir le même comportement syntaxique

- exemple : sélectionner une préposition pour le mot *janvier* => *en*

  on suppose que dans le corpus, on sait que:

  - *mai*, *juillet*, septembre sont souvent précédés de la préposition *en*
  - *janvier* est similaire à *mai*, *juillet*, *septembre* car ils sont utilisés dans des contextes similaires
    ex. *le mois de NOM-DE-MOIS*, *NOM-DE-MOIS prochain*

Généralisation de comportements syntaxiques - 2

- contexte :

  manger une (pomme + poire + orange)

- est-ce-que le verbe *manger* sélectionne *mangue* ?

  oui car *mangue*, *poire*, *pomme* et *orange* sont similaires !