Le Traitement automatique du langage naturel

CHLOÉ CLAVEL – TELECOM-PARISTECH

Plan du cours

- **Introduction au TALN**
- L'analyse de données textuelles – étiquetage morpho-syntaxique
- Les méthodes linguistiques
- Les méthodes statistiques – *machine learning*

TALN : Traitement Automatique du Langage Naturel

- Domaine à la frontière de:
  - L'intelligence artificielle
  - La linguistique
  - L'informatique
- Objectifs:
  - Compréhension du langage naturel : dériver du sens à partir de données textuelles
  - Générer automatiquement du langage
- En anglais : Natural Language Processing NLP

https://www.youtube.com/watch?v=Ea_ytYoUDso Luc Steels - BREAKING THE WALL TO LIVING ROBOTS. How Artificial Intelligence Research Tries to Build Intelligent Autonomous Systems

Les enjeux applicatifs du TAL

- La traduction automatique (Google translate)
- La fouille de données textuelles/Le classement des documents/L'extraction d'information
- Les correcteurs orthographiques
- Les résumés automatiques
- L'interaction humain-machine
- La reconnaissance de la parole
- La synthèse de la parole
- L'analyse des opinions sur le web social

Interaction humain-machine

- Kirobo : le robot japonais qui est parti 18 mois dans l'espace pour tenir compagnie à un astronaute

Les données et leurs enjeux

Enjeux :
des données qui s'éloignent de plus en plus du texte littéraire: des acronymes, des hashtags, des fautes de frappes, des fautes d'orthographes, etc.

Plan du cours

- Introduction au TALN
- **L'analyse de données textuelles – étiquetage morpho-syntaxique**
- Les méthodes linguistiques
- Les méthodes statistiques – *machine learning*

Les étapes préalables à l'analyse de données textuelles

1. Segmentation du texte en unités lexicales :
   - mots et phrases
2. Le traitement lexical :
   - déterminer les informations lexicales associées à chaque mot isolément (règles morphologiques et dictionnaire)
3. Le traitement syntaxique :
   - Désambigüiser en fonction du contexte, extraire les relations grammaticales que les mots et les groupes de mots entretiennent entre eux
     - Analyse morpho-syntaxique
     - Chunking

Ex: « Le compteur intelligent Linky sera déployé à Paris en 2013. »

1. Le/Compteur/Intelligent/Etc.
2. Le : déterminant masculin singulier ou pronom personnel masculin singulier
3. Le : déterminant masculin singulier

Exemple d'applications

- La synthèse vocale

Tests sous Acapella : les poules couvent au couvent.

- Première analyse pour la construction de règles linguistiques d'extraction d'information

*(manque|~negation-patt|(il/#NEG/y/avoir/~negation-patt))/(#PREP_DE)?/(conseil|contact|~services-lex)**

- Le prétraitement des données pour la classification de documents
  - Regrouper les formes fléchies des mots autour des lemmes (ex: infinitif pour un verbe, masculin singulier pour un nom)

Segmentation du texte en phrases et en mots

- Segmentation du texte en mots (tokenisation)
- Difficultés:
  - Gestion des balises, des marques et des variations typo-dispositionnelles (alinéas, tirets, blancs, tabulations…)
  - Détection de fin de phrase (localiser le point) : attention aux acronymes E.N.S.T., nombres (3.14), dates (29.05.2013)

Tests sous Acapella
Nous sommes le 16.05.2014. Il y a quelques années le nom de l'école était l'ENST ou mieux l'E.N.S.T.

L'étiquetage morpho-syntaxique

- Exemple :

Xerox

[10/1996, 10/1997]

La	le	+DET_SG
petite	petit	+ADJ2_SG
ferme	ferme	+NOUN_SG
du	de=le	+PREP_DE
père	père	+NOUN_SG
Fouchard	Fouchard	+NOUN_INV
se	se	+PC
trouvait	trouver	+VERB_P3SG
au sortir du	au sortir de=le	+PREP
défilé	défilé	+NOUN_SG
.	.	+SENT

Lemme

Catégorie lexicale

L'étiquetage morpho-syntaxique

- 2 types d'analyseurs syntaxiques
  - Analyseur déterministe: définition de règles (nécessite une grande expertise de la langue)
  - Analyseur probabiliste : apprentissage sur des corpus de données des probabilités de transitions entre catégories syntaxiques successives (nécessite des grands corpus de données)
- Difficultés :
  - compromis entre richesse de description et vitesse d'analyse
  - Gestion des ambiguïtés
  - Complexité des phénomènes à décrire
  - Robustesse aux entrées bruitées (coquilles, fautes de frappes, casse, etc.)

L’étiquetage morpho-syntaxique

- Les limites de l’analyse syntaxique – les ambiguïtés
  - La petite brise la glace ;

La désambiguïsation n’est possible qu’aux niveaux sémantique ou pragmatique

L'étiquetage morpho-syntaxique

- Analyseur déterministe : étiquetage par règle
  - Prise en compte du contexte local :
    - (4) DET/PRO V -> PRO V
  - Ex:
    - « Grammaires locales » d'INTEX (Silberztein, 1993)
    - Transducteurs de Xerox (Chanod & Tapanainen, 1995)
  - Méthode :
    - Implémentation (automates finis)
    - Fondement linguistique pour la construction des règles
  - Avantage : règles lisibles, modifiables manuellement, facilite la compréhension des erreurs
  - Inconvénient :
    - Écriture manuelle des règles : difficile, délicat et très coûteux.
    - Robustesse : traitement des entrées bruitées ; des mots hors vocabulaire

L'étiquetage morpho-syntaxique

- Etiqueteurs probabilistes : apprentissage sur des corpus de données des probabilités de transitions entre catégories syntaxiques successives (nécessite des grands corpus de données)
  - Très bons résultats
  - De nombreuses variantes (modèles MaxEnt, Champs Conditionnels Aléatoires)
  - Grand nombre de « règles probabilistes » (paramètres)
  - Fonctionnement en boîte noire
  - Plafonnement des performances, difficile de combiner des connaissances linguistiques

L'étiquetage morpho-syntaxique

- Formalisation du problème d'étiquetage
  - Modèle probabiliste sur les séquences de couples (mot, étiquette grammaticale) : p(M, E)

M =	·	·	·	w_{i-2}	w_{i-1}	w_i	·	·	mots
E =	·	·	·	e_{i-2}	e_{i-1}	e_i	·	·	étiquettes

  - Apprentissage/décision:
    - Apprentissage du modèle à partir d'un corpus étiqueté p(M, E)
    - Décision: la meilleure séquence E qui maximise le modèle p(M, E)
  - Autres exemples d'étiquetage séquentiel en TALN:
    - Reconnaissance d'entités nommées (cf cours F. Suchanek)
    - Analyse d'opinions

L'étiquetage morpho-syntaxique

- Etiqueteurs probabilistes : HMM
- Caractérisation du modèle $\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$
  - Le nombre *N* d'états du modèle (ici le nombre de catégories grammaticales)
  - Le nombre *M* de symboles distincts d'observation par état (ici le nombre de mots du vocabulaire)
  - La matrice **A** de transition entre états

$$a_{ij} = P[q_t = j | q_{t-1} = i] \qquad \text{pour} \quad 1 \leq i,j \leq N$$

  - La distribution **B** de probabilité d'observation des symboles dans l'état j

$$b_j(k) = P[\mathbf{o}_t = \mathbf{v}_k | q_t = j], \qquad \text{pour} \quad 1 \leq k \leq M$$

  - La distribution **Π** de l'état initial

$$\pi_j = P[q_1 = j], \qquad \text{pour} \quad 1 \leq j \leq N$$

L'étiquetage morpho-syntaxique

- Etiqueteurs probabilistes : HMM
  - Hypothèses simplificatrices :
    - Les suites d'étiquettes sont Markoviennes d'ordre k :

$$p(e_1 \ldots e_n) = p(e_1) \prod_{i=1}^{k} p(e_i|e_{i-1}) \text{ (à l'ordre 2)}$$

    - Conditionnellement aux étiquettes, les mots sont indépendants :

$$p(w_i|e_1 \ldots e_i, w_1 \ldots w_i) = p(w_i|e_i)$$

L'étiquetage morpho-syntaxique

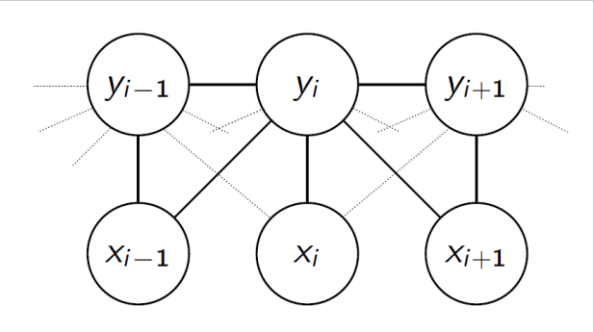
- Étiqueteur probabiliste : HMM
  - Les 3 problèmes des HMM
    - Problème 1 : évaluer la probabilité p(O/ λ) d'une séquence d'observations O en fonction du modèle λ = (**A, B, Π**) (algorithme forward-backward)
    - Problème 2 : retrouver la séquence d'états optimale connaissant la séquence d'observation O et le modèle λ = (**A, B, Π**) (algorithme de Viterbi)
    - Problème 3 : ré-estimer les paramètres du modèle λ = (**A, B, Π**) - ajuster les paramètres du modèle

Analyse syntaxique et HMM

- EXO :
  - On veut développer un étiqueteur morphosyntaxique reposant sur les modèles de Markov cachés à partir d'un corpus d'apprentissage constitué de textes annotés en N catégories grammaticales distinctes et contenant un vocabulaire de taille M.
    - Q1 : Présentez les 2 étapes nécessaires au développement de l'étiqueteur
    - Q2 : Préciser le problème des HMM correspondant à l'apprentissage du modèle (A, B, Pi) et préciser les différents éléments du modèle
    - Q 3 : Préciser le problème des HMM correspondant au problème suivant : comment étiqueter un nouveau texte en catégories grammaticales ?

Etiqueteur probabiliste : les CRF – les Champs Aléatoires Conditionnels

- généralisation des modèles de Markov cachés



- permettent d'intégrer via leurs fonctions caractéristiques des connaissances de nature très diverse.

$$F_j(\underline{x}, \underline{y}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, \underline{x})$$

- Le modèle appris par un CRF présente également l'avantage d'être relativement propice à l'interprétation :
  - l'importance d'une fonction caractéristique dans le modèle est caractérisée par son poids θ

$$p(\underline{y}|\underline{\mathbf{x}}; \boldsymbol{\theta}) = \frac{1}{Z(\underline{\mathbf{x}}, \boldsymbol{\theta})} \exp \sum_{j=1}^{D} \theta_j F_j(\underline{\mathbf{x}}, \underline{y})$$

  - permet d'identifier les connaissances qui jouent un rôle dans la tâche d'étiquetage

Etiqueteur probabiliste : les CRF – les Champs Aléatoires Conditionnels

- Les boîtes à outils existantes pour les CRFs
  - CRF suite http://www.chokkan.org/software/crfsuite/ et son wrapper python http://python-crfsuite.readthedocs.org/en/latest/
  - Wapiti : https://wapiti.limsi.fr/

Chunking (tronçonnage)

- Chunking
  - Détection des constituants syntaxiques :
    - Groupe nominal, noyau verbal, etc.
    - Repérage des frontières
    - Étiquetage par groupes (par l'étiquette de la tête)
  - John talked [to the children][about drugs]
- 2 approches :
  - Approches Symboliques : Spécification des patrons de groupes:
    - Ex: GN (Groupe Nominal) : DET ADJ* NN ADJ*
    - Implémentés par des transducteurs finis
  - Approches numériques :
    - Tâche d'étiquetage séquentiel (même technique pour l'étiquetage morpho-syntaxique)

Les outils existants pour l'étiquetage morphosyntaxique

- pour le français
  - Treetagger
  - Xerox, Brill [Brill, 1995]
  - LIA_Tag, macaon http://macaon.lif.univ-mrs.fr/index.php?page=home-en
- Pour l'anglais:
  - NLTK : http://www.nltk.org/
  - Treetagger

Plan du cours

- Introduction au TALN
- L'analyse de données textuelles – étiquetage morpho-syntaxique
- **Les méthodes linguistiques**
- Les méthodes statistiques – *machine learning*

Les méthodes linguistiques

- Objectif :
  - décrire l'information à extraire pour un métier, un domaine spécifique ou une thématique en modélisant l'information sous forme de lexiques/ontologies et patrons/règles linguistiques/grammaires/automates.

« manque de qualité de service »

« il n'y a vraiment pas eu de contact », …

→ Concept **INSATISFACTION**

Les méthodes linguistiques

- Modélisation sémantique :
  - Utilisation de lexiques et de règles
  - Règles qui répertorient toutes les formulations possibles d'une même information
    - langage d'expressions régulières
      - Appel de lemmes : ex. *« avoir »*
      - Appel de catégories grammaticales : *« #PREP_DE » « #NEG »*
      - Appel de lexiques prédéfinis: *« ~services-lex »*

« manque de qualité de service »

« il n'y a vraiment pas eu de contact », …

→ Concept **INSATISFACTION**

*(manque|~negation-patt|(il/#NEG/y/avoir/~negation-patt))/(#PREP_DE)?/(conseil|contact|~services-lex)**

* Exemple : syntaxe de l'outil TEMIS et exemple d'utilisation à EDF pour des analyses des opinions des clients

Les expressions régulières

- Syntaxe courante (Unix, perl, etc.)

Expression	Langage accepté
`r*`	0 ou plusieurs `r`
`r+`	1 ou plusieurs `r`
`r?`	0 ou 1 `r`
`[abc]`	`a` ou `b` ou `c`
`[a-z]`	N'importe quel caractère dans l'intervalle `a...z`
`.`	N'importe quel caractère sauf `\n`
`[^s]`	N'importe quel caractère sauf ceux de `s`
`r{m,n}`	Entre `m` et `n` occurences de `r`
`r1 r2`	La concaténation de `r1` et `r2`

Expression	Langage accepté
`r1 | r2`	`r1` ou `r2`
`(r)`	`r`
`^r`	`r` en début de ligne
`r$`	`r` en fin de ligne
`"s"`	Le string `s`
`\c`	Le caractère `c`
`r1 / r2`	`r1` quand il est suivi de `r2`

- `[a-zA-z]` Une lettre.
- `[0-9]` Un chiffre.
- `a[^A-Za-z]b` Un `a`, suivi d'un caractère non alphabétique, suivi d'un `b`.
- `^Monsieur` `Monsieur` en début de ligne.
- `[a-zA-Z]([a-zA-Z]|[0-9])*` Un identifiant Pascal. . . .

Tiré de http://www.ulb.ac.be/di/ssd/ggeeraer/lg/extexpreg_print.pdf

Les expressions régulières - exercice

- Donnez l'expression régulière acceptant l'ensemble des phrases «correctes» selon les critères suivants :
  - Le premier mot de la phrase a une majuscule ;
  - la phrase se termine par un point ;
  - la phrase est composée d'un ou plusieurs mots (caractères a…z et A…Z), séparés par un espace ;

Test des regexp :
http://www.regexplanet.com/advanced/java/index.html

Tiré de http://www.ulb.ac.be/di/ssd/ggeeraer/lg/extexpreg_print.pdf

Les expressions régulières - exercice

Donnez l'expression régulière acceptant l'ensemble des phrases «correctes» selon les critères suivants :
Le premier mot de la phrase a une majuscule ;
la phrase se termine par un point ;
la phrase est composée d'un ou plusieurs mots (caractères a...z et A...Z), séparés par un espace ;

Expression	Langage accepté
`r*`	0 ou plusieurs `r`
`r+`	1 ou plusieurs `r`
`r?`	0 ou 1 `r`
`[abc]`	`a` ou `b` ou `c`
`[a-z]`	N'importe quel caractère dans l'intervalle `a...z`
`.`	N'importe quel caractère sauf `\n`
`[^s]`	N'importe quel caractère sauf ceux de `s`
`r{m,n}`	Entre `m` et `n` occurences de `r`
`r1 r2`	La concaténation de `r1` et `r2`

Expression	Langage accepté
`r1 | r2`	`r1` ou `r2`
`(r)`	`r`
`^r`	`r` en début de ligne
`r$`	`r` en fin de ligne
`"s"`	Le string `s`
`\c`	Le caractère `c`
`r1 / r2`	`r1` quand il est suivi de `r2`

- `[a-zA-z]` Une lettre.
- `[0-9]` Un chiffre.
- `a[^A-Za-z]b` Un `a`, suivi d'un caractère non alphabétique, suivi d'un `b`.
- `^Monsieur` `Monsieur` en début de ligne.
- `[a-zA-Z]([a-zA-Z]|[0-9])*` Un identifiant Pascal. . . .

Tiré de http://www.ulb.ac.be/di/ssd/ggeeraer/lg/extexpreg_print.pdf

Les expressions régulières - exercice

- Donnez l'expression régulière acceptant l'ensemble des phrases «correctes» selon les critères suivants :
  - le premier mot de la phrase a une majuscule ;
  - la phrase se termine par un point ;
  - la phrase est composée d'un ou plusieurs mots (caractères a…z et A…Z), séparés par un espace ;

Sites pour vérifier les expressions régulières: regexplanet.com

^[A-Z][A-Za-z]*(\ [A-Za-z]+)*\.$

Tiré de http://www.ulb.ac.be/di/ssd/ggeeraer/lg/extexpreg_print.pdf

Les outils du marché

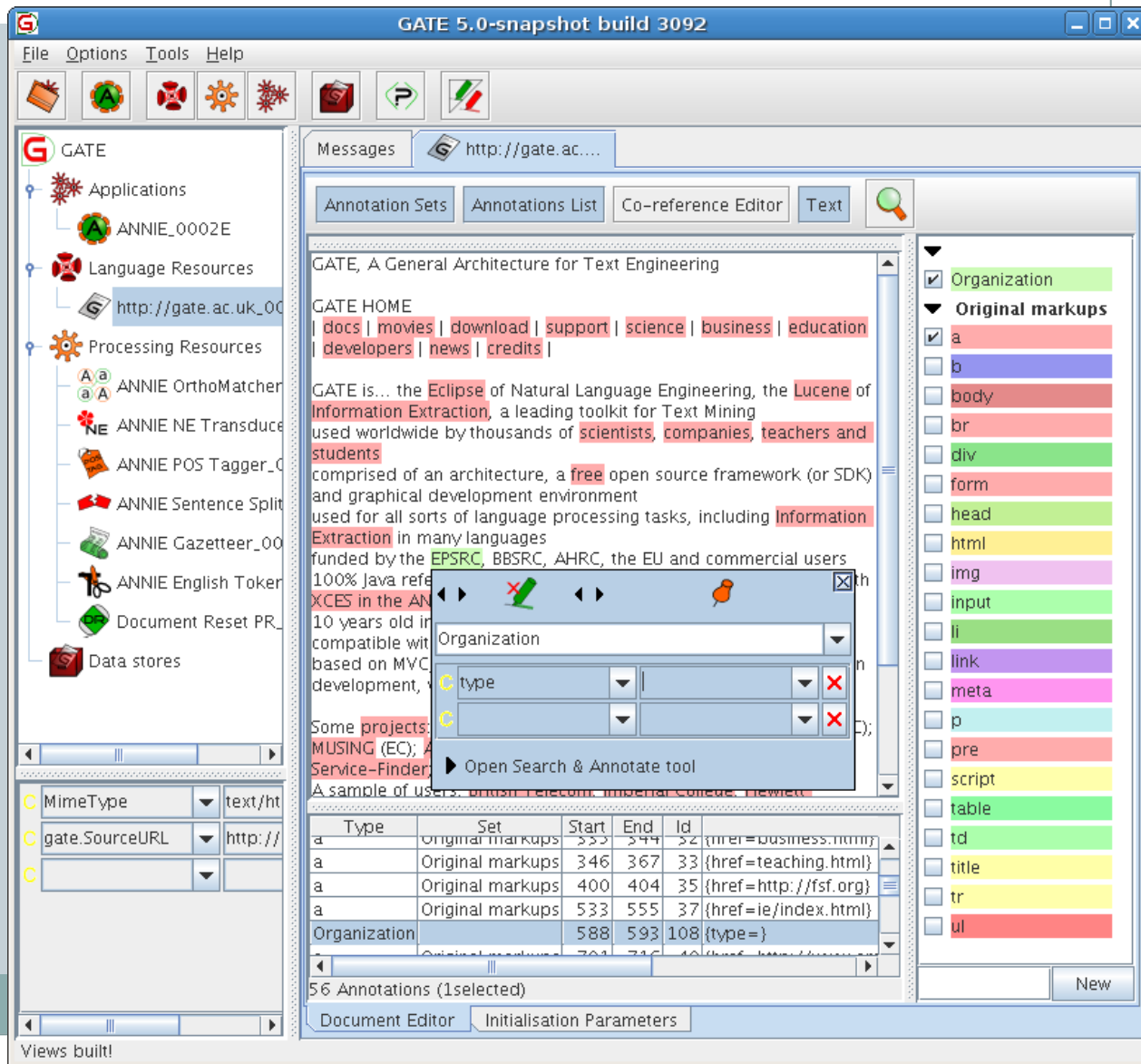
- Les outils qui permettent de modéliser les expressions
  - L'outil SemioLabs de la société Noopsis (voir autre cours)
  - L'outil LUXID de la société TEMIS
  - L'outil de la société Synapse

Les outils libres

- Unitex : http://www-igm.univ-mlv.fr/~unitex/
- Les grammaires de NLTK
- Gate

GATE

- General Architecture for Text Engineering,
- Suite Java pour l'extraction d'info et le NLP,
- Utilisé à l'échelle internationale avec des mises à jour continues,
- Intégration facile des différents outils et formats: divers taggers etc.



GATE

Fonctionnalités:

Système d'extraction d'information (ANNIE)

Annotation à base de règles: JAPE

Ontologies

Machine Learning

Dictionnaires externes (Gazetteer)

Permet une conception d'un système hybride: à base de règles + Machine Learning

- Interface pour l'annotation manuelle
- Possibilité d'intégrer GATE à Hadoop :

Hadoop-GATE https://github.com/wpm/Hadoop-GATE

GATE

- Différents exemples de projets de recherche avec GATE
  - Environnement web permettant d'effectuer les tâches d'annotation manuelle (crowdsourcing) (Bontcheva et al., 2014)
  - Interface permettant d'interroger des ontologies (Damljanovic, 2010)
  - Classification de textes en sentiments:
    - GATE+SVM (Funk, 2008)
    - À base de règles JAPE

GATE : JAPE Grammars

- Voir le tutoriel : https://gate.ac.uk/sale/thaker-jape-tutorial/GATE%20JAPE%20manual.pdf
- Exemple :
  - Texte : *AC Milan player David Beckham is going to comment on his future with LA Galaxy, who are eager to keep him in USA.*
  - Règle : If mention of the word “player” followed by a name of a person Then the person = a player.

```
Phase:nestedpatternphase
Input: Lookup Token
//note that we are using Lookup and Token both inside our rules.
Options: control = brill
Rule: playerid
(
 {Token.string == "player"}
)
:temp
(
{Lookup.majorType == Person}
|
(
 {Token.kind==word, Token.category==NNP,
Token.orth==upperInitial}
 {Token.kind==word, Token.category==NNP,
Token.orth==upperInitial}
)
)
:player
-->
 :player.Player= {rule = "playerid"}
```

Plan du cours

- Introduction au TALN
- L'analyse de données textuelles – étiquetage morpho-syntaxique
- Les méthodes linguistiques
- **Les méthodes statistiques – *machine learning***

Machine learning

- Types de tâches:
  - Classer, catégoriser les documents en thèmes, en opinions, etc.
    - La catégorisation ou classification supervisée
    - Le clustering ou classification non supervisée
  - Repérer des expressions
    - Ex: détection d'entités nommées

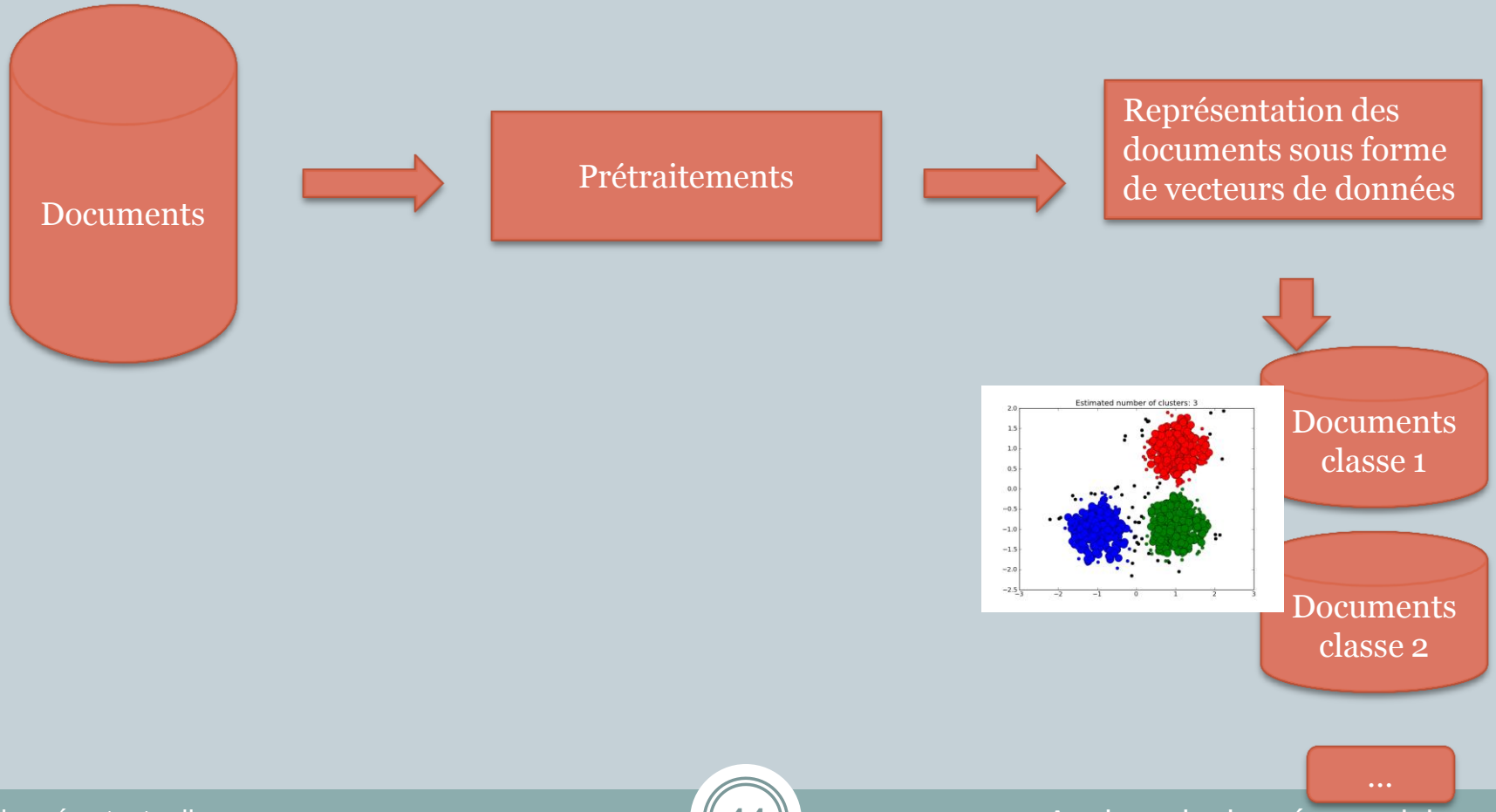
Tirée de http://www.tal.univ-paris3.fr/plurital/travaux-2009-2010/bao-2009-2010/MarjorieSeizou-AxelCourt/webservices.html

Catégorisation – les deux phases

- Phase 1 – l'apprentissage
  - Corpus d'apprentissage = ensemble de documents textuels annotés,
    - Annotation : chaque document est associé à une classe :
      - Ex1. Corpus d'articles de journaux : le thème de l'article (international, politique, sciences, sports, etc).
      - Ex2. Corpus de critiques de films : la note donnée par l'internaute (1 à 5)
  - Objectif : Apprendre à partir des données du corpus les caractéristiques communes à chaque classe
- Phase 2 – le test/la classification/la décision
  - À chaque nouveau document en entrée du système est attribuée automatiquement une classe

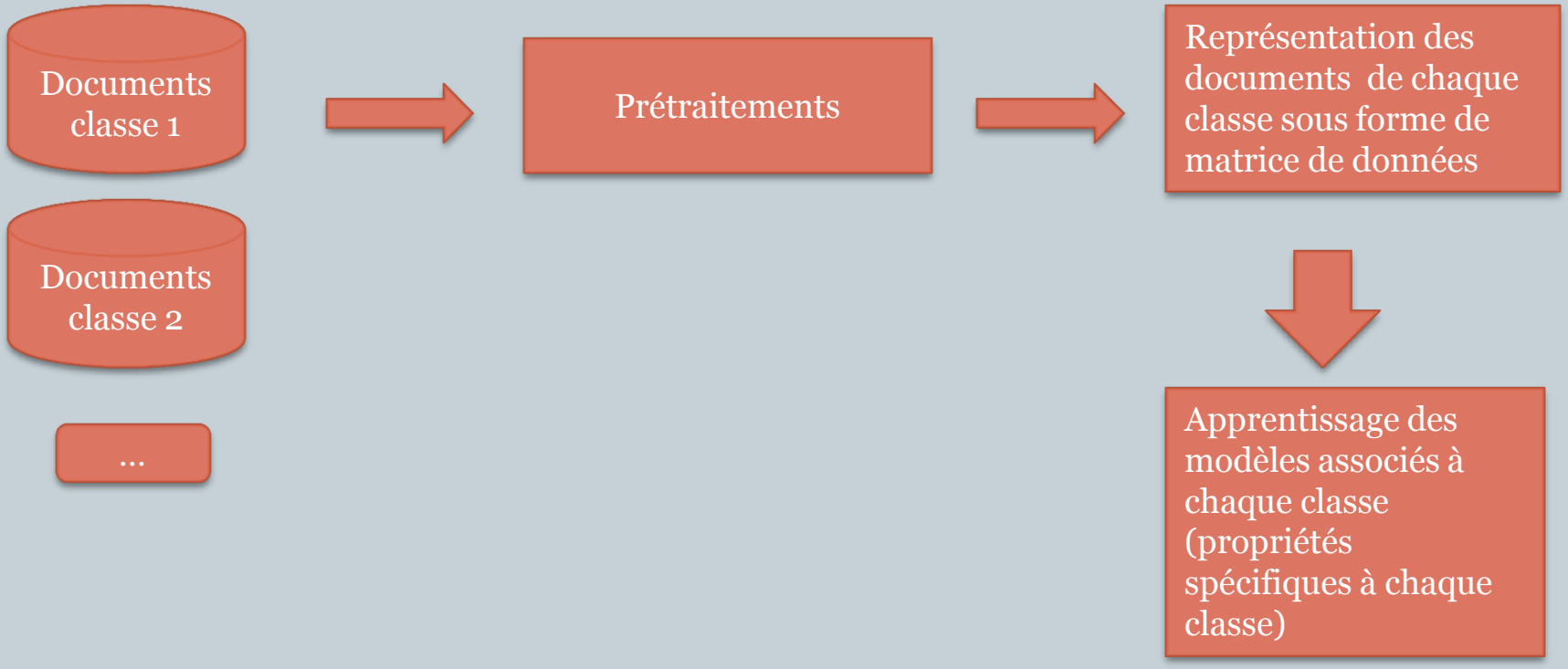
Clustering de documents

- Classification non supervisée

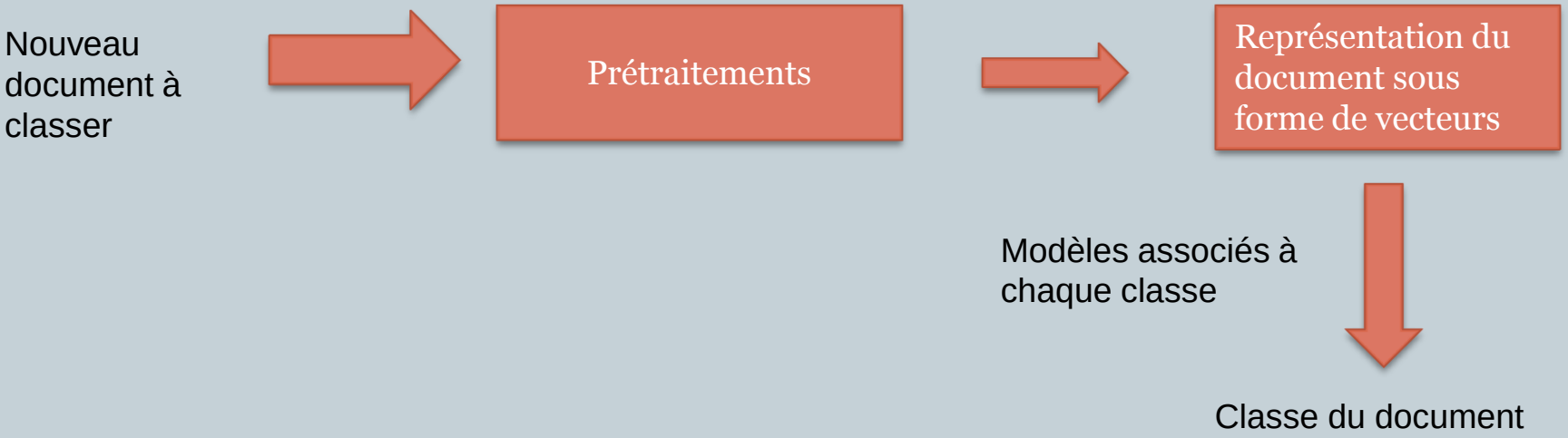


Catégorisation – phase 1 : l'apprentissage

- Apprentissage des classes



Catégorisation - phase 2 :la décision



Prétraitements

- Segmentation en mots / tokenization : choix des mots à considérer
  - Filtrage des signes (ponctuation, dates)
  - Filtrage des anti-mots (stop words) à partir d'une liste de mots
    - Mots de liaisons et d'articulation du texte car peu de pouvoir discriminant
  - Filtrage des hapax
    - termes qui sont très peu fréquents dans le corpus
    - Peuvent correspondre à des mots mal orthographiés
  - Regrouper des termes autour de leur racine ou de leur lemme
    - Racinisation (stemming) : tronquer certains suffixes
    - Lemmatisation (après une analyse morphosyntaxique)
  - Grouper les mots en n-grammes
    - Ex: considérer tous les couples de mots (bigrammes)
    - Ex: regrouper les termes appartenant au même syntagme

FORM	TAG
I	#PronPers
Would	#Verb
Like	#Verb
More	#QuantCmp
Contacts	#Noun
With	#Prep
EDF	#ProperName

Représentation du document sous forme de matrices de données

- 1 doc = 1 vecteur (a1, …., aN) de longueur N (le nombre de mots dans l'ensemble des textes)
  - où a_i = nombre d'occurrences du mot i dans le texte
  - où a_i = TFIDF du mot i dans le texte
    - TFIDF (Term Frequency Inverse Document Frequency) = mesure statistique utilisée pour évaluer la représentativité d'un terme/mot par rapport à un document dans une collection de textes
    - La représentativité du terme augmente proportionnellement au nombre de fois où le terme apparaît dans un document (TF), mais il est pondéré par sa fréquence dans l'ensemble du corpus (IDF)
- Base de documents = matrices terme/document

Calcul de TF-IDF

- Formule TF-IDF du mot w dans le document d

$$TFIDF(w,d) = TF_{w,d}.IDF_{w,d}$$
$$= TF_{w,d}.\left(\left(\log_2 \frac{N}{DF_w}\right)\right)$$

  - N : le nombre total de documents dans le corpus
  - TF : Term Frequency
    - nombre d'occurrences de w dans le document considéré (on parle de « fréquence » par abus de langage).
    - Variantes :
      - fréquences booléennes: tf(w,d) = 1 si w dans d, 0 sinon
      - logarithmically scaled frequency: tf(w,d) = 1 + log f(w,d), ou 0 si f(w, d) est 0;
  - DF : Document Frequency
    - nombre de documents contenant le mot w
- Exercice
  - Ex 1 : La base contient 1000 documents, calculer la TF-IDF du mot « compteur » dans le document d, sachant que le document d contient 3 fois le mot compteur et que 70 textes contiennent également le mot « compteur »

Calcul TF-IDF

- Exercice 1
  - Ex 1 : La base contient 1000 documents, calculer la TF-IDF du mot « compteur » dans le document d, sachant que le document d contient 3 fois le mot compteur et que 70 textes contiennent également le mot « compteur »
  - TF-IDF(« compteur », d) = $3.\left(\log_2 \frac{1000}{70}\right) = 11{,}5$
- Exercice 2
  - Le mot « compteur » apparaît toujours 3 fois dans le document mais apparait cette fois dans 900 documents

Calcul TF-IDF

- Exercice 2
  - Le mot « compteur » apparaît toujours 3 fois dans le document d mais apparait cette fois dans 900 documents
  - TF-IDF (« compteur », d) = $3.\left(\log_2 \frac{1000}{900}\right) = 0.45$

=> Le poids du mot compteur dans le document est moins important

Classification non supervisée

- Exemples de méthodes
  - K-moyennes
    - Principe général
      - documents = points d'un espace multi-dimensionnel, muni d'une distance d.
      - Initialisation: Les documents sont dans un premier temps aléatoirement affectés à chaque classe 1…K. + Calcul du centroïde de chaque classe comme barycentre des individus du groupe:

$$\mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} x_i, \quad \forall k \in \{1, \ldots, K\}$$

      - Itération: calcul de l'inertie, critère d'arrêt = convergence de l'inertie

$$\sum_{k \in \{1,\ldots,K\}} \sum_{i \in \mathcal{C}_k} \|x_i - \mu_k\|_2^2$$

Choix de la distance?

Classification non supervisée

- Choix de la distance/mesure de similarité pour les k-means
  - Métrique la plus courante en texte: similarité cosinus
    - Similarité entre 2 vecteurs de doc A et B en fonction du cosinus de l'angle

$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

  - Autre mesure de similarité, l'indice de Jaccard $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$
    - La distance associée $J_\delta(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$

Classification non supervisée

- Exemples de méthodes
  - Mélange de lois multinomiales:
    - initialisation :
      - considérer un ensemble K de clusters et initialiser les paramètres de la loi associée à chaque cluster
      - Attribuer chaque document à un cluster en fonction de la probabilité du document d'appartenir à une classe (classe la plus probable) -> partitionnement initial
    - itération :
      - Recalculer les paramètres du modèle sur la base des clusters du partitionnement courant
      - Redistribuer les documents dans les clusters à partir de ce nouveau modèle.

Classification non supervisée

- Exemples de méthodes:
  - Analyse sémantique latente, analyse en composantes principales, analyse des correspondances
    - Principe:
      - décomposition de matrices selon leurs directions propres (ou singulières) pour conserver un maximum d'information sur un nombre minimum de dimensions.
      - La décomposition en valeurs singulières de la matrice terme/document permet d'obtenir des thèmes dominants dans le corpus, chacun étant associé à un sous-espace singulier.
    - Outil pour l'analyse sémantique latente : http://lsa.colorado.edu/
    - Alternative pour les problématiques d'indexation :
      - indexation sémantique latente [Deerwester et al 90] : LSI
        - fournir la réponse à une requête en regardant les similarités entre la requête et les documents.
        - considérer les liens sous-jacents entre des termes dans le corpus (découvrir une structure latente dans le corpus).
        - utiliser la LSI permet de gérer les problèmes de polysémie, synonymie et hyperonymie(terme de la requête est vêtement et le terme du document est robe).

Exemple de méthodes de classification supervisée

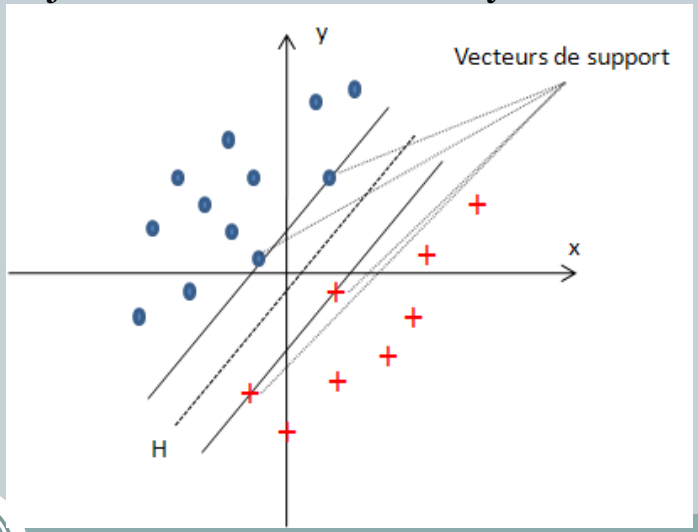
- Le classifieur Bayésien naïf (Naive Bayes Classifier)

Exemple de méthodes de classification supervisée

- Les k-plus proches voisins
  - Méthode non bayésienne et non paramétrique d'apprentissage supervisé
    - Avantage : pas de règle de décision de type bayésien, pas d'hypothèses sur les lois de probabilité, pas d'estimation des paramètres des lois
    - Inconvénient: coûteuse en temps de calcul, adaptée si beaucoup d'exemples d'apprentissage
  - Principe général
    - Classement du point inconnu en fonction de la classe de ses voisins dans l'ensemble d'apprentissage
    - Apprentissage : connaître la classe des éléments de l'ensemble d'apprentissage
  - Algorithme de décision :
    - Pour chaque document testé, calculer sa similarité avec les documents du corpus d'apprentissage
      - Calcul des similarités entre le vecteur de mots du document et les vecteurs de mots de tous les documents de l'apprentissage.
    - Récupération des k vecteurs les plus proches du document testé (k similarités les plus élevés)
    - Décision : classe du document = classe majoritairement attribuée aux k documents étudiés.

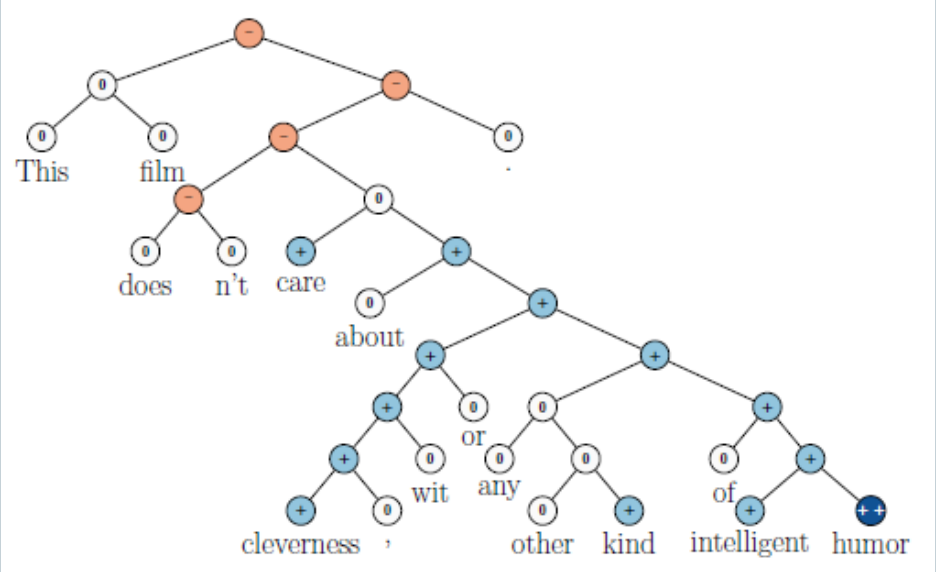
Exemples de méthodes de classification

- Les SVM – Support Vector Machines [Vapnik, 1995]
  - Principe général de l'apprentissage
    - Séparer les exemples d'apprentissage de chaque classe en maximisant la distance à l'hyperplan
    - Vecteurs supports : les points les plus proches de l'hyperplan
    - Marge : distance minimale entre l'hyperplan et les exemples d'apprentissage
    - => Apprentissage = maximiser la marge
  - En pratique
    - projeter les données dans un espace de dimension plus grand afin de se ramener à un problème linéaire via une fonction de projection dite fonction noyau
    - Exemples de noyau:
      - Linéaire : $k(x,y) = x.y$
      - Gaussien $k(x,y) = \exp(\frac{-\|x-y\|^2}{2\sigma^2})$
      - Polynômial $K(x,y) = (1 + x.y)^d$
  - Décision : calcul de la position du nouvel exemple par rapport à l'hyperplan



Réseaux de neurones et deep learning

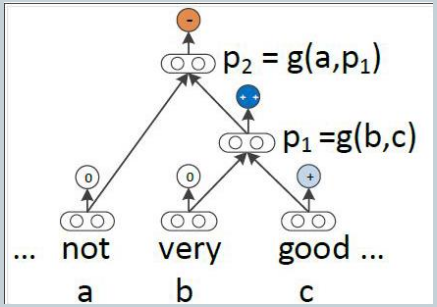
- Remise au goût du jour des réseaux de neurones avec l'émergence du deep learning
  - Utilisation des réseaux récursifs tensoriels
    - permettent de prendre en compte la structure d'une phrase.



- exemple d'utilisation des réseaux récurrents
  - REF : R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA : Association for Computational Linguistics, October 2013, pp. 1631? 1642.

Réseaux de neurones et deep learning

- Utilisation des réseaux récursifs tensoriels
  - Représentation de la phrase par un arbre (utilisation du parseur de Stanford)
  - On applique les récursivement les fonctions d'activation:



  - Apprentissage : apprentissage de la fonction g du passage au parent dans l'arbre binaire de représentation la phrase

Les méthodes d'étiquetage séquentiel pour des tâches d'extraction d'information

[Localité d'Ukraine] menace les livraisons de gaz à l' UE
. affaire Madoff contient encore de nombreuses zones d
de l' UE sous l'il de **Paris** [Communes de France] . La
tionnisme de **Nicolas Sarkozy** [Chef d'État] . Avec l'
ement culturel . La **Russie** [Pays] a cessé de fournir
ent] n' a pas à craindre pour ses approvisionnements .
le de l' occupation américaine en **Irak** [Pays] . Le
ourées entre jeunes et policiers . Des engins incendiaires

- même méthodes que celles qui sont utilisées pour l'étiquetage morpho-syntaxique : CRF et HMM.
- Exemple de tâche: détection d'entités nommées (voir cours Fabian Suchanek), outils sur étagère (apprentissage à base de CRF):
  - pour le français
    - LIA_NE http://pageperso.lif.univ-mrs.fr/~frederic.bechet/download.html (appris sur des données issues de l'oral)
    - SEM http://www.lattice.cnrs.fr/sites/itellier/SEM.html (appris sur des données écrites, des phrases tirées du journal Le Monde)
  - pour l'anglais:
    - l'étiqueteur d'entités nommées de Stanford appris sur des données variées (CoNLL, MUC-6, MUC-7 and ACE) http://nlp.stanford.edu/software/CRF-NER.shtml

Détection d'entités nommées

- Les données annotées selon le modèle BIO

```
     Wolff B-PER
         , O
 currently O
         a O
journalist O
        in O
 Argentina B-LOC
         , O
    played O
      with O
       Del B-PER
    Bosque I-PER
        in O
       the O
     final O
     years O
        of O
       the O
 seventies O
        in O
      Real B-ORG
    Madrid I-ORG
         . O
```

Représentation des mots sous forme de vecteurs sémantiques

- Objectif : fournir une représentation des mots sous forme de vecteurs qui capturent les relations sémantiques entre les mots
- Exemple d'outil : word2vec de Google https://code.google.com/p/word2vec/
  - Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
  - Technique inspirée du deep-learning qui permet d'améliorer les performances des méthodes de classification de documents (incluant la classif d'opinions) ou d'extraction d'information.
- Voir aussi le papier de Stanford:
  - Maas, Andrew L., et al. "Learning word vectors for sentiment analysis."*Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011.
  - Technique inspirée de la LDA : Latent Dirichlet Allocation (modèle de topic probabilistique)

	romance	romance	romance
	love	charming	screwball
romantic	sweet	delightful	grant
	beautiful	sweet	comedies
	relationship	chemistry	comedy

Quelques pointeurs

- Outils de classification :
  - NLTK : modules python open source pour le TAL et scikitlearn http://nltk.org/ et http://scikit-learn.org/
  - Weka : plateforme java permettant d'expérimenter facilement les classifieurs et algorithmes d'apprentissage: http://www.cs.waikato.ac.nz/ml/weka/

Exemple d’applications

- Systèmes de recommandation, upselling

Pour aller plus loin

QUELQUES RÉFÉRENCES

Références du TAL

- Cours :
  - *Une petite introduction au traitement automatique des langues naturelles* par François Yvon http://perso.limsi.fr/Individu/anne/coursM2R/intro.pdf
  - *Introduction au TALN et à l'ingénierie linguistique* par Isabelle Tellier http://www.lattice.cnrs.fr/sites/itellier/poly_info_ling/info-ling.pdf
- Etiqueteur morpho-syntaxique
  - [Brill, 1995] *Transformation-based error-driven learning and natural language processing : A case study in part-of-speech tagging*. Computational Linguistics, 21(4), 543–565.
- Traitement de l'oral :
  - [Blanche-Benveniste, et al., 1990] *Approches de la langue parlée en français,* Claire Blanche-Benveniste, L'essentiel francais, Orphrys