

Traitement automatique des langues probabiliste

Carlos Ramisch



Langage Naturel

Plan

- 1 Introduction
- 2 Traitement de corpus
- 3 Statistique textuelle

Plan

① Introduction

- Motivations et applications

- Lexiques

- Corpus

- Modèles

② Traitement de corpus

③ Statistique textuelle

Le domaine

Traitement automatique des langues (TAL)

Science pluridisciplinaire à l'intersection entre



- *informatique*
- *linguistique*
- *sciences cognitives*

qui vise à modéliser les *langues humaines* dans les *systèmes informatiques*, pour permettre des interactions en langue naturelle homme-machine et entre humains



Les langues humaines sont. . .

Omniprésentes !

- Interactions parlées entre personnes (et systèmes informatiques)
- Popularité des applications web et mobiles →   . . .
- Très grand volume de texte généré tous les jours
- Naturellement multilingue



Mais les langues humaines sont aussi...

- En constante évolution :
 - *allô quoi !*
 - *printemps arabe*
 - *big data*
 - *LOL, mdr, ptdr, ...*
- Ambiguës :
 - *vers (de terre) × vers (préposition) × vert × verre*
 - *il était tard × il est têtard*
 - *avocat (fruit) × avocat (métier)*
- Arbitraires :
 - *pleine lune × ?lune entière*
 - *lait entier × ?lait complet*
 - *pain complet × ?pain plein*



Quelques applications de TAL

- Traduction automatique
- Recherche d'informations
- Synthèse et reconnaissance vocale
- Recherche d'informations
- Systèmes de question-réponse
- Apprentissage de langues (étrangères)
- Agents conversationnels - chatbots
- Inférence textuelle
- Simplification
- *Business intelligence* - fouille de textes
- ...

Données non structurées

- Objet d'étude principal du TAL : **le texte**
- Suite d'octets \implies information non structurée
- Problématique du TAL : trouver la *structure* dans le texte pour pouvoir la traiter par l'informatique
- Dans quel contexte faut-il traiter des données textuelles ?
- Quelle est la bonne représentation pour cette structure ?

Fondamentaux du TAL

Les systèmes de TAL manipulent :

- Des dictionnaires (lexiques)
- Des textes (corpus)
- Des grammaires, graphes, etc. (modèles)

Ressources lexicales

- Au cœur des applications de TAL
- Contiennent des informations sur les *unités lexicales*
- Données structurées, plus qu'une liste de mots
- Lexiques : dictionnaires, terminologies, thésaurus, ontologies, ...



- Éléments fondamentaux d'une langue
- Blocs de LEGO
- Ont un sens (\neq affixes ou mots fonctionnels)
- Morphologie ne change pas l'unité lexicale (lexème)
- Exemples : *souris*, *machine à laver*, *faire part*



Un exemple : Princeton's WordNet

- Unités lexicales groupées en **synsets** (par sens)
- Macro-structure : monolingue, Anglais, liens inter-synset et inter-mot : synonymie, antonymie, hyperonymie, etc.
- Micro-structure : définitions, parties du discours, exemples, fréquence, etc.

WordNet en ligne

<http://wordnet.princeton.edu/>

Comment on construit un lexique ? I

Approche standard

- des années de travail
- des dizaines de lexicographes expérimentés
- des milliers d'€
- pour des humains, pour des ordinateurs (ou les deux) ?
- très haute qualité



Comment on construit un lexique ? II

L'approche "paresseuse"

- apprendre des lexiques automatiquement à partir du texte
- indépendant de la langue
- sale, rapide et pas cher
- requiert grands volumes de texte et ordinateurs puissants
- bruit et silence



Qu'est-ce qu'un corpus ?

- Une collection **représentative** de textes pour une étude (Lettres, Histoire, etc)
- Une **grande** collection de textes (Informatique)

C'est-à-dire

Une collection de textes (suffisamment grande) utilisée dans une tâche linguistique (étude théorique, construction d'une application...)

Corpus monolingue, parallèle, comparable

- **Monolingue** : une langue
- Multilingue :
 - **Parallèle** : les textes dans les 2 langues sont des traductions, alignées au niveau de la phrase
 - **Comparable** : les textes sont similaires (domaine, registre) mais pas des traductions

Corpus annotés

Couche d'annotation manuelle sur le texte permet d'appliquer des techniques d'apprentissage automatique

- Parties du discours, arbres syntaxiques, anaphores, termes, sens des mots, rôles thématiques, actes de parole, etc
- Automatique (bruit) ou humaine (lente et chère)
- Accord inter-annotateur
- Standards de représentation et format absents



Domaine, registre, modalité, équilibre

- **Domaine** : knowledge area of the texts
 - Corpus spécialisés : industrie aéronautique, pédiatrie, cardiologie, loi environnementale, etc.
- **Registre (genre)** : rôle/contexte social du texte
 - Journaux, blogs, twitter, littérature, modes d'emploi
- **Modalité** : écrit ou oral (transcriptions)
- Un corpus **générique** est **équilibré** sur plusieurs domaines, registres et modalités

Exemples de corpus

Corpus	langue	domaine	registre
French Treebank	monolingue	générique	news
Penn Treebank	monolingue	générique	news
GENIA	monolingue	génétique	scientifique
British National Corpus	monolingue	générique	équilibré
FrWaC	monolingue	générique	web
Wikipedia dumps	comparable	générique	encyclo.
Europarl	parallèle	politique	oral non spontané
TED Talks	parallèle	générique	oral non spontané
Bible	parallèle	sacré	littérature

Où peut-on trouver des corpus ?

- Télécharger gratuitement, demander à l'auteur, *corpora* list
 - <http://dumps.wikimedia.org/>
 - <http://www.statmt.org/europarl/>
 - <http://opus.lingfil.uu.se/>
 - <http://wacky.sslmit.unibo.it/>
 - ...
- Faire un dump du web
 - wget-it-yourself
 - BootCat <http://bootcat.sslmit.unibo.it/>
- Acheter auprès de LDC <https://www.ldc.upenn.edu/> ou ELRA
<http://www.elra.info/>
- Construire manuellement

Modèles pour le TAL

- Symboliques
 - Automates finis (morphologie)
 - Dictionnaires électroniques
 - Grammaires et arbres de dérivation
 - Grammaires logiques et prédicats
- **Probabilistes**
 - “Sacs de mots” (classifieurs)
 - Modèles de séquence (n-grammes, HMM)
 - Modèles de graphes (parseur)
 - Modèles de traduction

TAL probabiliste

Constat : Il est fastidieux, voire impossible, d'écrire des règles contextuelles et lister toutes les exceptions possibles pour tous les phénomènes linguistiques de toutes les langues

Proposition : Au lieu d'écrire les règles, on peut demander à l'ordinateur de les **apprendre** à partir de données où ces phénomènes apparaissent (textes).

- Problème classique en intelligence artificielle
- Apprentissage automatique - machine learning
- Probabilités - désambiguisation

Apprentissage automatique

- Construire un système de TAL
 - Expertise linguistique/informatique
 - Apprentissage supervisé
 - Apprentissage non supervisé
- Que peut-on apprendre automatiquement à partir des textes ?
- De quoi a t-on besoin pour apprendre en TAL ?

Méthodologie TAL probabiliste

Recette pour développer un système de TAL par apprentissage automatique à partir de corpus

- 1 Définir une tâche
- 2 En déduire un guide d'annotation
- 3 Collecter des données
- 4 Demander à des gens d'annoter ces données
- 5 Créer un système automatique pour faire la tâche
- 6 Évaluer la qualité du système

Exemple TAL probabiliste

Classification thématique de textes

1 Définir une tâche

Classification en **thèmes** : sports vs politique

2 En déduire un guide d'annotation

Choisir l'étiquette en fonction du thème abordé par le texte, si le texte est ambigu, l'écarter

3 Collecter des données

Collecter des pages web sur sport.fr et assemblée-nationale.fr

4 Demander à des gens d'annoter ces données

C'est déjà fait

5 Créer un système automatique pour faire la tâche

Si on trouve "joueur", "match", "gagner", "perdre", c'est du sport, sinon c'est de la politique

6 Évaluer la qualité du système

Calculer le nombre de fois où le système s'est trompé sur les données collectées, en déduire un **taux d'erreur**

Outline

① Introduction

② Traitement de corpus

Traitements sur corpus

Rappel regexp

③ Statistique textuelle

Traitements sur corpus

Avant d'être exploitable, un corpus doit être pré-traité

- Formats de fichiers : html, xml, txt, pdf
- Structure des documents : pages, paragraphes, titres
- Structure des phrases : segmentation, tokenisation
- Caractères : encodage, majuscules
- Nettoyages : longueur des phrases, dates, URLs, listes, équations, etc.

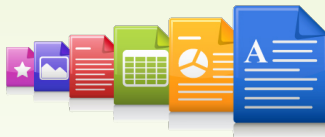
Traitements sur corpus

Avant d'être exploitable, un corpus doit être pré-traité

- Formats de fichier : html, xml, txt, pdf
- Structure des documents : pages, paragraphes, titres
- Structure des phrases : segmentation, tokenisation
- Caractères : encodage, majuscules
- Nettoyages : longueur des phrases, dates, URLs, listes, équations, etc.

Fichiers de texte

- Extraire le texte des .pdf, .html, .docx
- Fichiers .txt “Structurés”
- Chronophage



Des octets aux caractères

- Encodage de caractères : un grand cauchemar
- ASCII : les débuts de l'informatique
- Unicode code-point <http://www.unicode.org/>
 - UTF-8, UTF-16LE, UTF16BE
- Windows (latin1) et Unix (utf-8)
commandes iconv et file
- Fin de ligne (CR, LF, CR+LF)
Commandes fromdos et todos

Des caractères aux phrases

- La plupart des outils de TAL utilisent les phrases comme unités (parsing, TA)
- Simple : une phrase finit par un point
- Complexe : acronymes, slashes, dates, nombres, URLs, etc ...
- D'habitude, on utilise des *expressions régulières* et sed

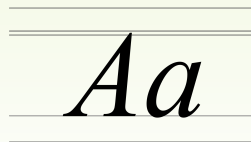
Des phrases aux mots

- Le processus de découpage de la phrase en mots est appelé **tokenisation**
- Simple : la plupart des langues utilisent des espaces (et le Chinois ?)
- Complexe : tirets, ponctuation, contractions (*du*), elision (*l'*)...
- Expressions régulières, listes d'exceptions, systèmes probabilistes



Majuscules et minuscules

- Parfois on doit homogénéiser les majuscules et minuscules
 - *The Table is RED. A red table broke.*
- Parfois non...
 - *so I talked to Mr. Smith about ABBA.*
- Stratégie la plus utilisée : lowercase
- Attention aux corpus spécialisés
- Stratégies d'homogénéisation plus "intelligentes"



Nettoyage

- Dépend du corpus et du domaine
- Usage de caractères spéciaux pour la mise en page
- Exemples d'heuristiques :
 - Phrases trop courtes/longues
 - Proportion de caractères non alphabétiques (smileys, etc.)
 - Proportion de mots-outil
 - Présence de verbes
 - Boilerplate



Expressions régulières

Outils de base en TAL pour chercher (grep) et transformer (sed, awk) des fichiers texte

- **a** : caractère a minuscule
- **.** : un caractère quelconque
- **?** : Le caractère précédent apparaît 0 ou 1 fois (optionnel)
- ***** : Le caractère précédent apparaît 0 ou plusieurs fois
- **+** : Le caractère précédent apparaît 1 ou plusieurs fois
- **[a-z]** : un caractère entre a et z
- **[^a]** : un caractère quelconque différent de a
- **^** : début de ligne
- **\$** : fin de ligne

Outils de la ligne de commande

- **cat** : Afficher le contenu d'un fichier
- **head/tail** : Afficher les premières/dernières lignes
- **less** : Afficher le contenu d'un fichier avec déroulement
- **sort** : Trier des fichiers
- **uniq** : Éliminer des doublons sur des fichiers triés
- **uniq -c** : Éliminer des doublons + les compter
- **cut** : extraire des colonnes de fichiers CSV
- **paste** : coller des colonnes
- **join** : jointure de colonnes

Utiliser **man** + forums en ligne

Langage awk

- Langage de programmation avec syntaxe similaire au C
- Permet de manipuler des colonnes d'un fichier texte via les variables spéciales \$1 \$2 \$3 etc
- Appliquer les modifications sur les lignes qui correspondent à une regexp
- Variables spéciales FS, OFS, NF, NR, etc.

```
/^#{/
{
    for(i=1;i<=NF;i++){
        total += $i;
        print "Total jusqu'à présent : " total;
    }
}
END{
    print "Total final: " total;
}
```

Plan

① Introduction

② Traitement de corpus

③ Statistique textuelle

Rappel : probabilités

Compter des mots et des n -grammes

La loi de Zipf

Rappel : probabilité

Chance/risque qu'un événement a arrive :

- $P(a)$: probabilité **a priori** de l'événement a . $P(a) \in [0, 1]$
- $P(a|b)$: probabilité **conditionnelle** de a sachant b . Chance que a arrive lorsque b est arrivé.
- $P(a, b)$: probabilité **jointe** de a et b . Chance que les deux événements arrivent conjointement. Si a et b sont **indépendants**, $P(a, b) = P(a) \times P(b)$ (a n'a pas d'influence sur b et réciproquement). Si a a une influence sur b , $P(a, b) = P(a) \times P(b|a)$.

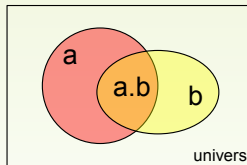
Estimation de probabilités

- Fréquentielle (maximum de vraisemblance) :

$$P(a) = \frac{\text{nombre de } a}{\text{nombre d'événements}}$$

$$P(a, b) = \frac{\text{nombre de } a \text{ avec } b}{\text{nombre d'événements}}$$

$$P(a|b) = \frac{\text{nombre de } a \text{ avec } b}{\text{nombre de } b}$$



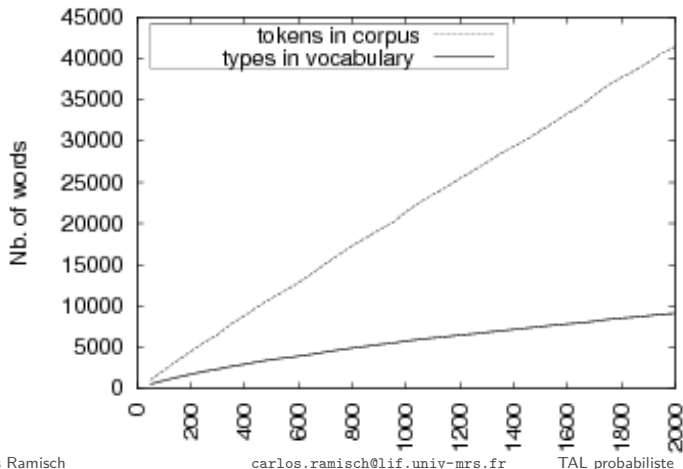
Tokens et types

- Les **tokens** sont les unités minimales dans un corpus
- Le **vocabulaire** (**types**) est l'ensemble de **tokens distincts** dans le corpus
- Chaque occurrence d'un type est un token distinct
- Chaque type apparaît une seule fois en tant que mot vedette dans un dictionnaire
- Un token peut apparaître plusieurs fois dans un corpus
- Type-token ratio → richesse du vocabulaire

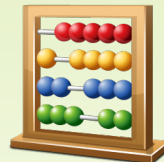
$$\text{TTR} = \frac{|\text{vocabulary}|}{|\text{corpus}|}$$

TTR

La valeur de TTR varie avec la taille du corpus car le nombre de tokens croît linéairement alors que le nombre de types croît logarithmiquement.



Loi de Zipf



- Les nombres d'occurrences des tokens dans un corpus suivent la *loi de Zipf*
- Autres noms : loi de puissance, distribution de Paretto, LNRE (large number of rare events)
- Le nombre d'occurrences $c(w_i)$ d'un token w_i est inversement proportionnel à son rang $r(w_i)$.

$$c(w_i) \approx \frac{1}{r(w_i)^a}$$

- a est un paramètre qui dépend du corpus

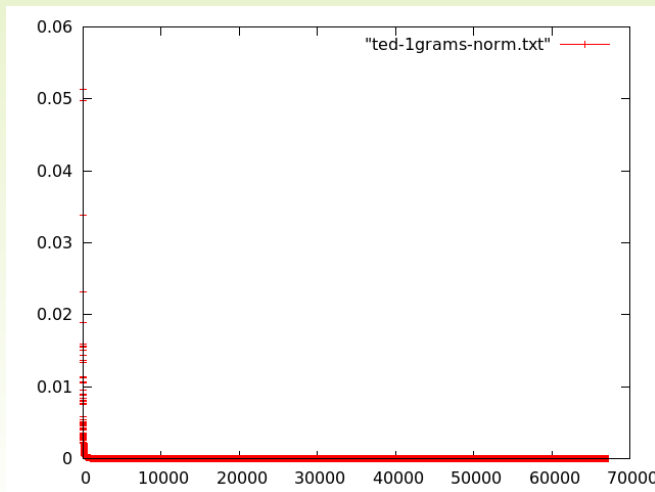
La loi de Zipf hante le TAL

Conséquences de la loi de Zipf

- La plupart des types apparaît une seule fois
- Les statistiques obtenues pour les mots rares ne sont pas fiables
- Peu importe la taille du corpus, certains mots n'y apparaîtront pas
- Biais vers les mots très fréquents
- Effet combinatoire - groupes de tokens



Histogramme des fréquences I



Histogramme des fréquences II

