Introduction à la donnée linguistique et au traitement du langage naturel (NLP)

G. Durantin

ISAE-Supaéro

Introduction

Propriétés remarquables du langage

Problématique

- Qu'est ce qui distingue le langage des autres systèmes de communication ? (animaux, informatiques...)
- En quoi le langage nécessite-t-il un traitement particulier ?

L'infinité discrète

- A partir d'un nombre fini de **symboles**, on produit une infinité de phrases, de longueur non limitée.
- Symboles ou événements linguistiques = sons, mots, caractères, gestes...

Conséquences pour le traitement du langage

A partir d'un signal aux valeurs possibles limitées, il existe une infinité de manières d'exprimer une idée $\Longrightarrow$ Complexité de traitement


Browse


Eye makeup


Frown


Unattention


Bite


Hair barrette

D'autres animaux, comme les primates non humains par exemple, utilisent également des symboles pour communiquer. La manière de combiner les informations n'est en revanche pas infinie dans ce qu'on observe chez ces espèces

Le déplacement

- Le langage permet de décrire des éléments concrets, mais également des éléments abstraits ou absents (éloignés géographiquement ou temporellement)

Conséquences pour le traitement du langage

L'objet du discours n'est par conséquent pas forcément accessible autrement que par le langage (puisqu'il est abstrait) $\Longrightarrow$ Ne pas traiter suffisamment le langage, c'est se priver d'informations autrement inaccessibles

Chez les abeilles, les patterns de danse peuvent décrire des objets loin dans l'espace. Mais ce langage n'inclut pas de notion d'abstrait ou de futur/passé

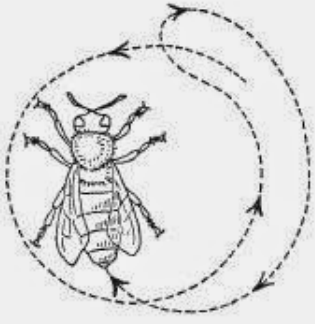


Figure 1.
Round dance

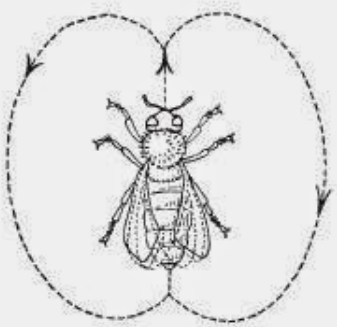
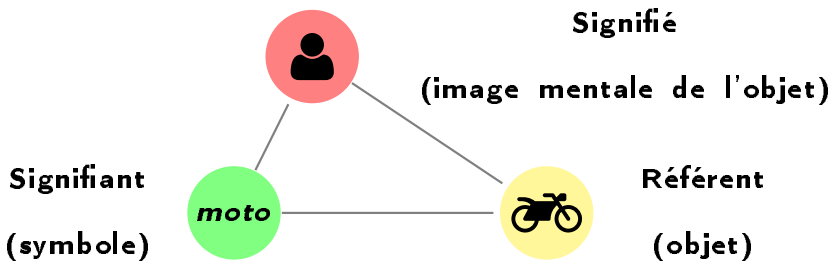


Figure 2.
Waggle dance

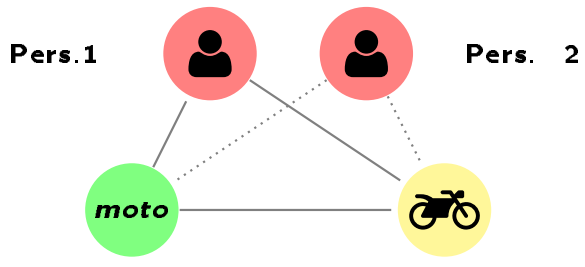
L'attention jointe

et l'intentionalité partagée

- Le langage est un **vecteur de coopération**, et par conséquent flexible



Triangle sémiotique de Pierce : mécanisme de "création" du sens



Configuration d'attention jointe : ajustement flexible des signifiants en présence des référents

Conséquences pour le traitement du langage

Les symboles utilisés pour faire référence à un objet ou un concept sont par conséquent très flexibles au cours des interactions $\Longrightarrow$ Les systèmes de traitement doivent aussi utiliser des représentations flexibles du langage

Chez certaines espèces d'oiseaux comme les corbeaux, les individus coopèrent régulièrement. Cette coopération ne repose toutefois pas sur l'utilisation de symboles (cris, gestes) flexibles mais plutôt sur des reflexes acquis

L'attachement à une culture

- La langue est attachée à une culture.
- La production du langage est conditionnée par de nombreux facteurs socio-culturels. Cette multiplication de facteurs fait qu'il est difificle de prédire le langage utilisé dans un cas précis.

Conséquences pour le traitement du langage

Le langage est soumis à de multiples facteurs socio-culturels $\Longrightarrow$ Les systèmes de traitement du langage seront moins performants lorsqu'ils seront prédictifs plutôt que descriptifs.

Rappel des propriétés remarquables

Infinité discrète

A partir d'un nombre fini de symboles, le langage produit par assemblage une infinité de messages

Déplacement

Le langage permet de décrire des objets ou concepts en dehors du "ici et maintenant"

Attention jointe

Le langage permet d'ajuster de manière flexible la communication, afin d'arriver à un but

Attachement à une culture

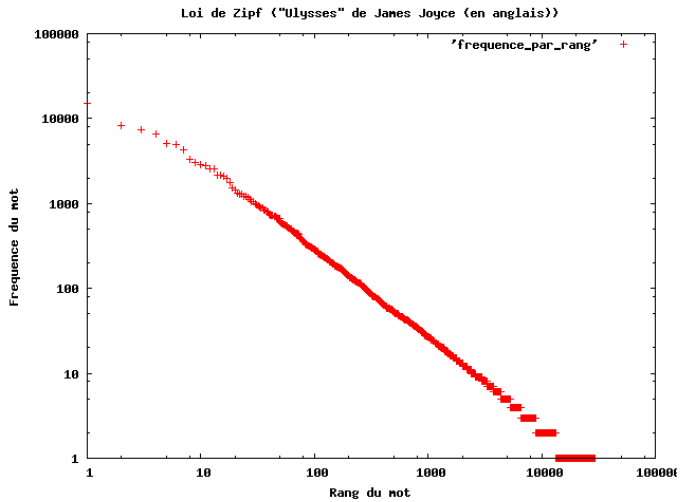
Le langage est un constituant de l'identité culturelle. La production du langage est associée à de nombreux facteurs socio-culturels.

Structures du langage

Problématique

- On a vu que le langage est constitué de symboles assemblés en messages, et que sa production est soumise à de multiples facteurs peu prédisibles.
- Peut-on toutefois en dégager une structure ?

Problématique



- Dès la première moitié du XX$^{\text{ème}}$ siècle, plusieurs auteurs (e.g. Zipf) constatent que la fréquence des événements linguistiques semble suivre un schéma structuré.
- Cette loi a été formalisée et démontrée par Mandelbrot, en confrontant deux éléments essentiels à la production du langage d'un point de vue psychologique : le coût de stockage du langage (dans le cerveau) et son coût d'utilisation (lorsque l'on veut produire le message)

Loi de Zipf-Mandelbrot

Démonstration

Loi statique de Shannon

Le coût de stockage de l'information est proportionnel au logarithme de la quantité d'information à encoder

$$c_{stockage}(n) = a + b.ln(cn + d) \tag{1}$$

Loi dynamique de Shannon

A l'optimalité, les symboles les moins fréquents sont affectés aux informations les plus coûteuses à encoder

Il y a donc décroissance de la fréquence avec le coût d'utilisation. [Mandelbrot, 1965] montre que de manière optimale :

$$f(n) = \kappa e^{-\lambda c_{utilisation}(n)} \tag{2}$$

Loi de Zipf-Mandelbrot

Démonstration

Hypothèse de Mandelbrot

Le coût d'utilisation d'un symbole est directement proportionnel à son coût de stockage

$$c_{stockage}(n) = \gamma c_{utilisation}(n) \tag{3}$$

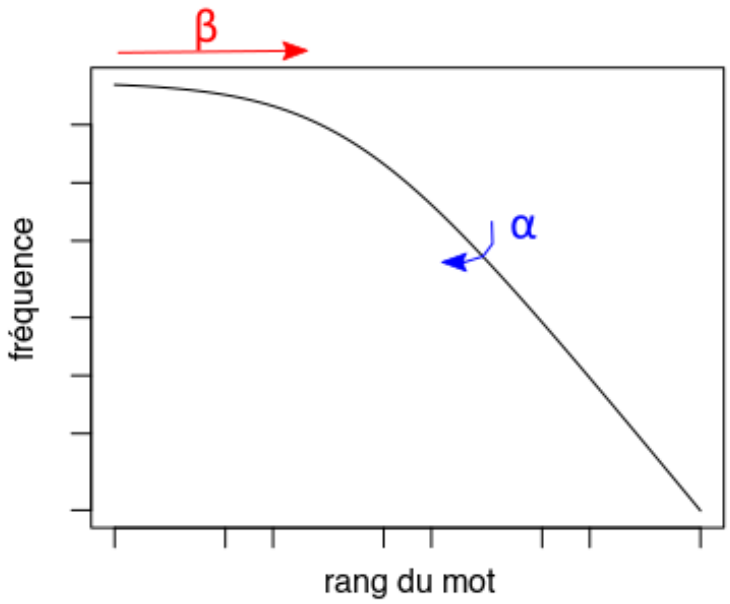
En combinant les équations 1, 2 et 3, et en posant $K = \kappa e^{-a\gamma\lambda} c^{-b\gamma\lambda}$, $\alpha = b\gamma\lambda$ et $\beta = d/c$, on obtient la **loi de Zipf-Mandelbrot** :

Loi de Zipf-Mandelbrot

$$f(n) = \frac{K}{(n+\beta)^{\alpha}} \tag{4}$$

Loi de Zipf-Mandelbrot

Forme



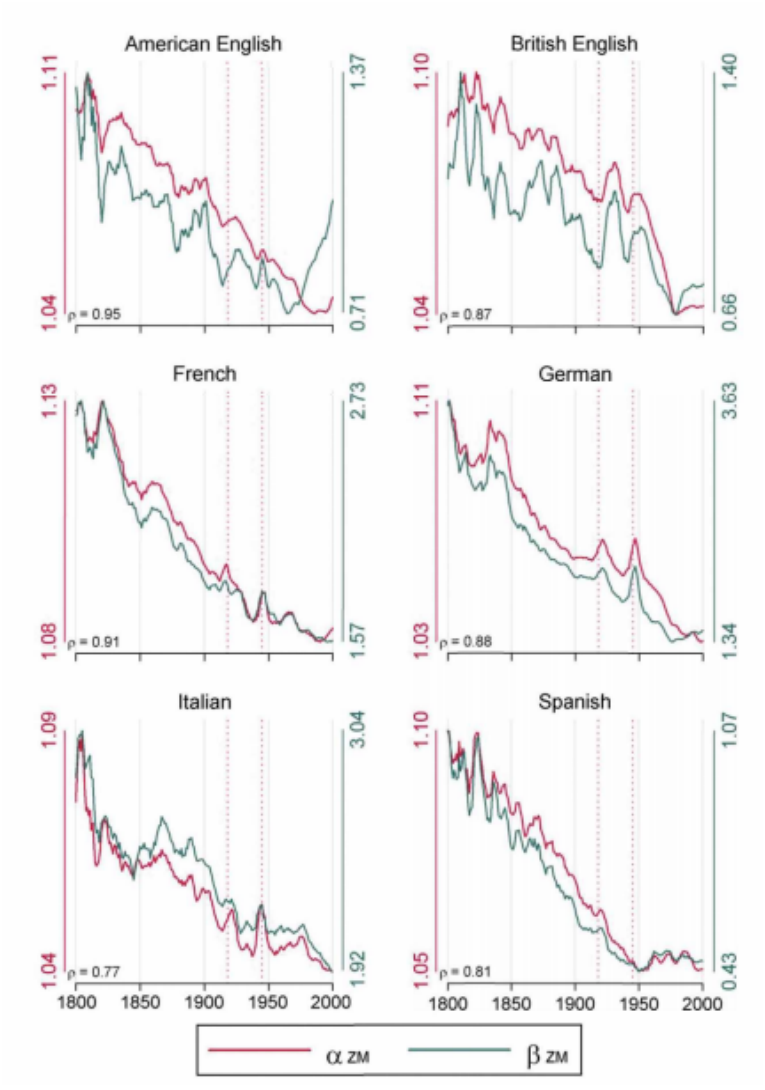
La courbe log-log typique de la loi de Zipf-Mandelbrot est linéaire décroissante et présente un "coude". Le paramètre α influe sur la pente de la courbe et le paramètre β influe sur le rang d'apparition du coude.

Pour $\beta = 0$, le coude disparaît et on obtient la loi empirique de Zipf, observée empiriquement dès 1912

Zipf-Mandelbrot dans les langues

- Typiquement, on constate que $\alpha \approx 1.1$ et $0 \lessapprox \beta \lessapprox 4$
- Ces paramètres varient toutefois fortement d'une langue à l'autre, en fonction du type de discours considéré, de facteurs socio-économiques ou temporels.

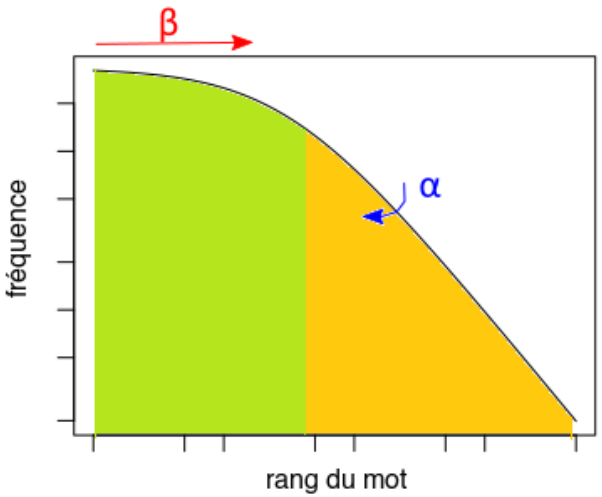
Zipf-Mandelbrot dans les langues : [Koplenig, 2018]



Problématique de Long Tail

- Vu cette répartition des événements linguistiques, peut-on espérer limiter le traitement de ces événements à un noyau réduit ?

Problématique de Long Tail : exemple fictif



Considérons un "vocabulaire" de 2000 événements linguistiques pouvant survenir. Ceux ci surviennent avec une loi de fréquence de Zipf-Mandelbrot $f_{ZM}(n, \alpha, \beta) = K(n + \beta)^{-\alpha}$, avec $\alpha = 1.1$ et $\beta = 1.6$ [1]

Considérons un système de traitement du langage optimisé pour traiter les κ événements les plus fréquents. Lors de l'utilisation de ce système, la probabilité qu'un événement linguistique tombe hors de ce noyau d'événements est alors donnée par :

$$p_{LT}(\kappa) = \frac{\sum_{i=\kappa+1}^{2000} f_{ZM}(i, \alpha, \beta)}{\sum_{i=1}^{\kappa} f_{ZM}(i, \alpha, \beta) + \sum_{i=\kappa+1}^{2000} f_{ZM}(i, \alpha, \beta)} \quad (5)$$

[1] Ces valeurs sont proches de celles estimées pour le français moderne écrit

Problématique de Long Tail : exemple fictif

- Cas pour $\kappa = 20$: $p_{LT}(\kappa) = 0.58$ (58% des cas tomberaient donc hors du périmètre du système)
- Cas pour $\kappa = 50$: $p_{LT}(\kappa) = 0.45$
- Cas pour $\kappa = 100$: $p_{LT}(\kappa) = 0.36$
- Cas pour $\kappa = 1500$: $p_{LT}(\kappa) = 0.08$

Problématique de Long Tail

Pris ensemble, les événements linguistiques minoritaires représentent une masse importante.

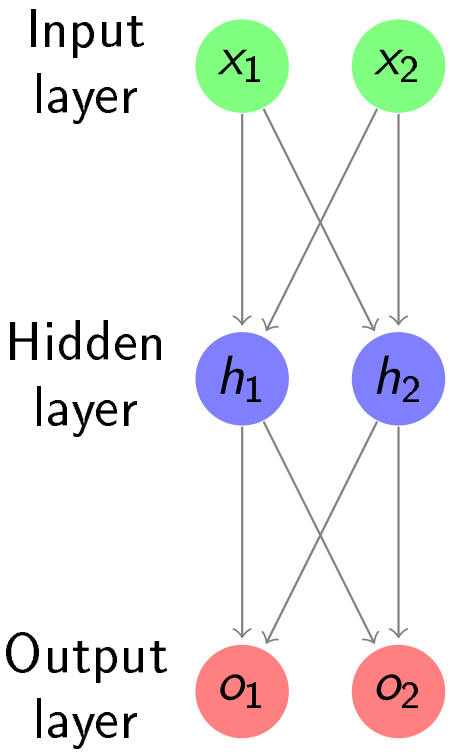
En conséquence de la problématique de Long Tail, les systèmes de traitement de la langue doivent faire preuve d'une grande flexibilité pour offrir des niveaux de traitement adéquats à tous les événements linguistiques. (cf section 1:Raisonnements symboliques et statistiques)

Problématique

- On a vu que le langage est constitué répond à certaines règles de structure, en particulier sur la fréquence d'apparition des mots.
- Au-delà des aspects fréquentistes, le langage possède-t-il une structure qu'il est possible de faire apprendre à une machine ?

Trouver de la structure dans le langage

Rappels sur la propagation du gradient dans un réseau de neurones



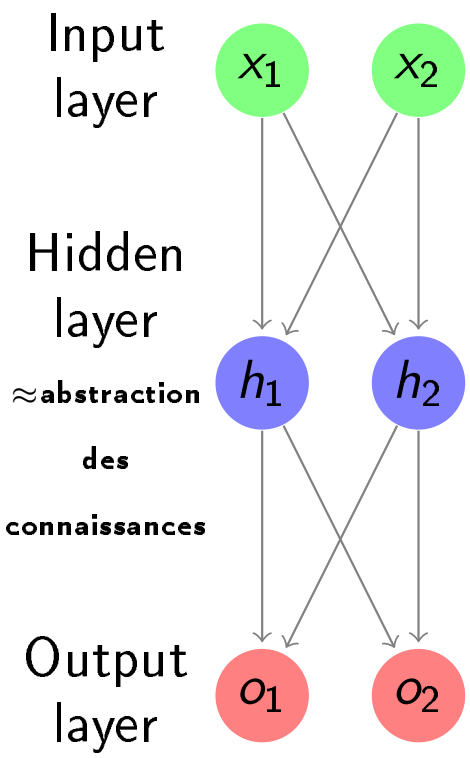
:

- Chaque flèche du réseau correspond à un poids (w_{ij}) liant un neurone (i) d'une couche à un neurone (j) d'une autre couche
- Lors de la propagation vers une couche inférieure (exemple hidden layer), on calcule la composante par combinaison linéaire des composantes de la couche supérieure : $h_j = \sum_{i=1}^{n} w_{ij} x_j$

Lors de la séance 2, on reverra la rétropropagation permettant d'ajuster les poids en fonction de l'erreur durant l'apprentissage

Trouver de la structure dans le langage

Le problème de prédiction de la lettre suivante



onceuponatimeaboyandagirl

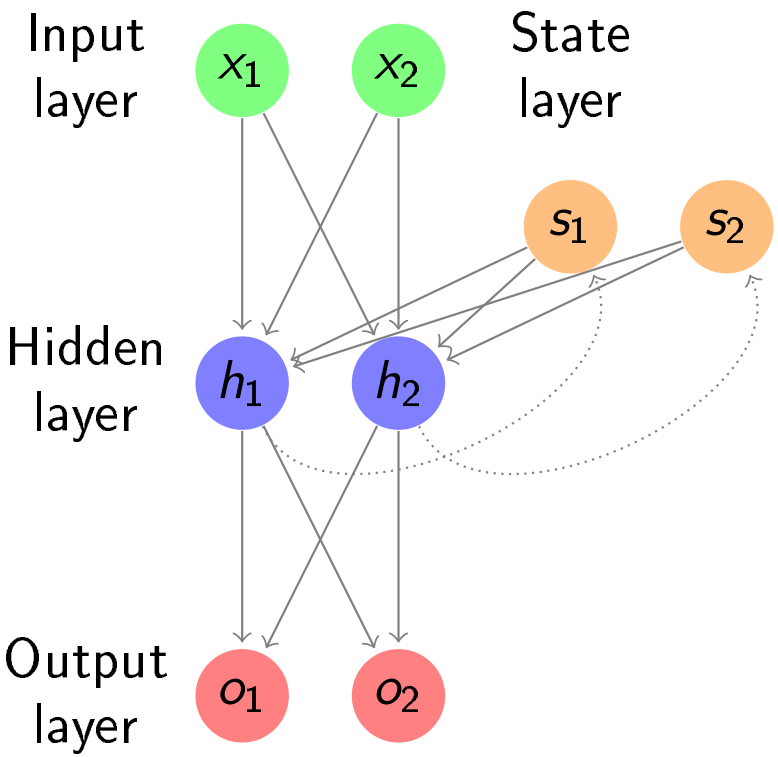
- input : lettre de la séquence (encodée sur 5 bits=5 neurones)
- output : lettre suivante dans la séquence (encodée de la même manière)

Le réseau de neurones présenté à gauche ne parvient pas à effectuer ce genre de tâche.

Même avec l'abstraction proposée par la *hidden layer*, la connaissance de la seule lettre en cours ne suffit pas pour prédire la suivante.

Trouver de la structure dans le langage

L'approche des réseaux d'Elman



[Elman, 1990] propose l'ajout d'un registre copiant à l'identique les activations de la *hidden layer* et servant d'entrée supplémentaire au système, pour conserver une mémoire de l'état du système. C'est la **State layer**[2]

[2] [Jordan, 1986] avait déjà proposé un raisonnement similaire, mais en utilisant un registre copiant la Output Layer. Elman constate toutefois que la copie d'instructions en mémoire sous forme abstraite -et donc via la Hidden Layer- semble mieux adaptée

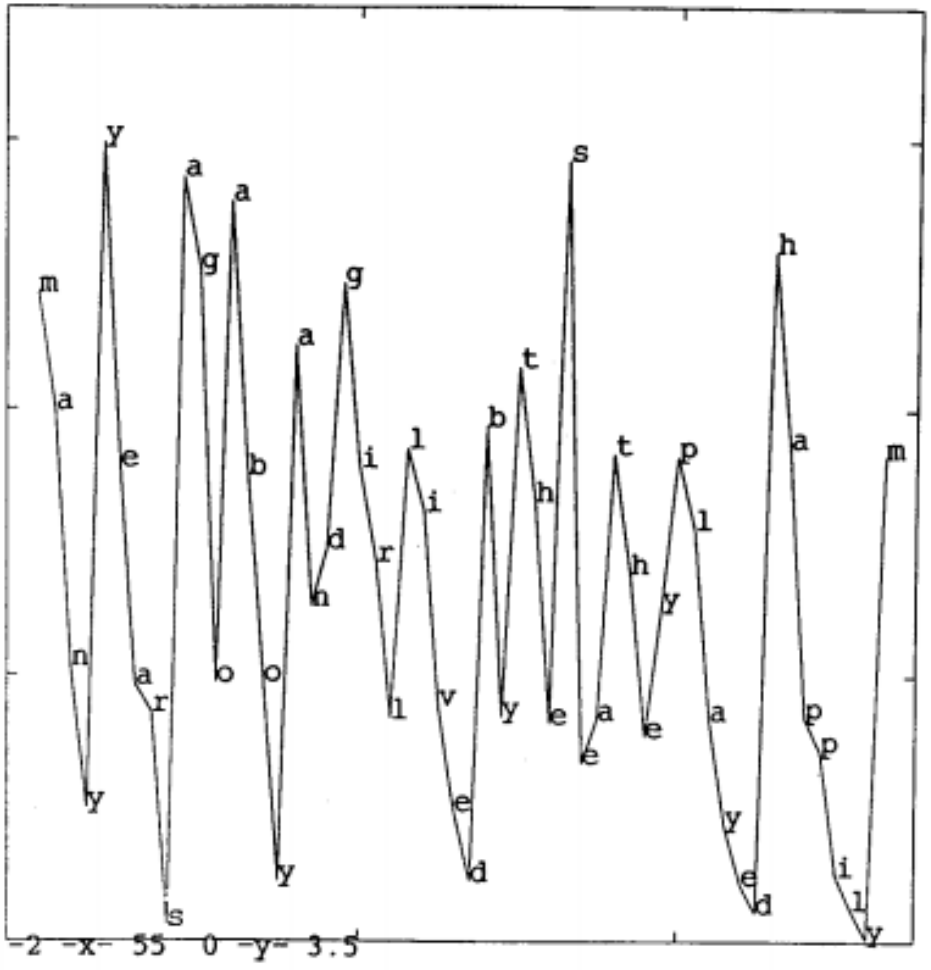


Figure 6. Graph of root mean squared error in letter-in-word precition task.

L'erreur de prédiction obtenue est forte en début de mot et diminue en fin.
Le réseau a appris à reconnaître les mots

Le langage possède donc une structure interne (non linéaire) qu'il est possible de faire apprendre à un réseau.

Les grandes familles de problèmes en NLP

Problèmes communs de bas-niveau

Ces problèmes sont des **tâches élémentaires de traitement du langage**. Elles sont en général combinées entre elles pour répondre à un besoin précis.

- Problèmes centrés sur un token
  - Sémantique lexicale ou distributionnelle : modélisation du sens du token
  - Part-of-speech (POS) tagging : labellisation du rôle grammatical du token (i.e. nom commun, verbe, pronom...)
  - Named Entity Recognition (NER) : labellisation des tokens appartenant à une classe définie (i.e. noms de pays, types de cuisine, noms de personnalités...)
- Problèmes centrés sur un groupe de tokens ou une phrase
  - Classification de texte
  - Analyse de sentiment ou de tonalité
  - Parsing en dépendances : attribution de relations grammaticales entre parties de la phrase
  - Extraction de relations : labellisation de liens de relation entre plusieurs tokens (e.g. détection des *aspects*, i.e. du sentiment associé à un token de la phrase)

Problèmes communs de haut niveau

Ces problèmes sont des **cas d'usages communs de traitement du langage**. Ils combinent généralement plusieurs approches de bas niveau pour y répondre.

- Natural Language Understanding (NLU)
  - Compréhension du sens d'un texte ou d'un message

    je voudrais réserver une table au restaurant chinois $\Longrightarrow$
    {intention:*réserver*; type_restaurant:*chinois*}
- Natural Language Generation (NLG)
  - Génération de langage
  - Inclut des cas d'usages comme la *Machine translation* (traduction) et la *Summarization* (résumé automatique)
- Question Answering (QA)
  - Association d'une question avec sa réponse

Modélisation du langage

Raisonnements symbolique et statistique en NLP

- Chaque *point de données* en NLP est un message contenant beaucoup d'information
- Pour faire face à la dimensionnalité et la complexité importante des données, de nombreuses approches en NLP passent par une abstraction intermédiaire

$$\textit{je voudrais réserver une table au restaurant chinois} \Longrightarrow \{\text{intention:}\textit{réserver}; \text{ type\_restaurant:}\textit{chinois}\}$$

- Cette abstraction permet de simplifier le traitement des messages, en faisant baser ce traitement sur la représentation **symbolique** du message.
- Ces approches sont appelées approches **symboliques**, par opposition aux approches **statistiques** qui se basent sur le message entier.

Exemple de raisonnement symbolique : les approches rule-based et les limites du raisonnement symbolique

Dans les approches par règles, le comportement du traitement est défini à l'aide de règles s'appliquant à des catégories de tokens (selon leur contenu, leur role grammatical, leur longueur, etc...).

EXAMPLE (tiré de la librairie Spacy)

```
# Matches "love cats" or "likes flowers"
pattern1 = [{"LEMMA": {"IN": ["like", "love"]}},
            {"POS": "NOUN"}]

# Matches tokens of length >= 10
pattern2 = [{"LENGTH": {">=": 10}}]
```

Limites

Rapides à mettre en place, les approches symboliques permettent de contrôler finement le comportement d'une solution
Toutefois, elles sont en général moins robustes que les approches statistiques, et ne fourniront une solution convenable que dans les cas les plus courants (orthographe parfaite, formulations très communes...)

Rappel : tokenisation

- Dans la définition des symboles (approches symboliques) ou la vectorisaton (approches statistiques), la plupart des approches font appel à la **tokenisation**
- Un token = unité élémentaire de composition du langage (ce n'est pas forcément un mot séparé des autres par un espace, e.g. *New York*, *arc-en-ciel*)

Je *prend* *l'* *avion* *pour* *New* *York* *aujourd'hui*

Dans la pratique, la tokenisation la plus simple peut être faite via séparation sur les espaces ou la ponctuation, mais peut aussi reposer sur des modèles plus complexes capables de reconnaître les structures élémentaires du langage (voir par exemple slide 25)

Rappel : n-grams

- Les modèles de **n-grams** reposent sur la prédiction de l'occurrence de séries de *n* tokens successifs dans la séquence.

Phrase d'origine	*je vais à New York*
Unigrammes	[***je***, ***vais***, ***à***, *New*, *York*]
Bigrammes	[*je vais*, *vais à*, *à New*, ***New York***]
Trigrammes	[*je vais à*, *vais à New*, *à New York*]

Les modèles n-grams permettent de représenter la structure des phrases et l'ordre des mots. Ils sont souvent utilisés pour améliorer la robustesse des approches basées sur des tokens. Ils augmentent cependant grandement la complexité des modèles, car ils multiplient le nombre de tokens

Rappels de vectorisation : le Bag-of-words

Dans cette représentation, un vocabulaire de K tokens est extrait du corpus de N documents (ou phrases) à vectoriser.
Chaque document est ensuite représenté en comptant le nombre d'occurrences de chaque token (noté *TF* pour *Term frequency*).

	Token 1	Token 2	...	Token K
Phrase 1	1	4	...	2
Phrase 2	3	0	...	5
...				
Phrase N	0	3	...	1

Limites du BOW

Dans le BOW, les mots très présents dans le corpus dans son ensemble (indifféremment de la phrase considérée) ont un poids important. Ils n'ont pourtant que peu de valeur pour différencier les documents.

Rappels de vectorisation : le TF-IDF

Soit $D = \{d_j\}$ l'ensemble des documents du corpus. A partir de ce corpus, on extrait $T = \{t_i\}$ le vocabulaire de tous les termes (tokens). On note alors tf_{ij} la fréquence du terme i dans le document j. Pour chaque terme, on définit également l'*inverse document frequency* $idf_i = log \frac{card(D)}{card(\{d_j | t_i \in d_j\})}$ (soit l'inverse du ratio des documents contenant le token)
On définit alors le TF-IDF par :

$$tfidf_{ij} = tf_{ij}.idf_i$$

Avantages du TF-IDF

Comparativement au BOW, le TF-IDF atténue l'importance des tokens qui seraient très présents dans l'ensemble des documents du corpus. Il met donc l'accent sur les mots les plus discriminants entre les documents, ce qui en fait une méthode de vectorisation **très adaptée aux problèmes de classification**

Limites des BOW et TF-IDF

- Les BOW et TF-IDF reposent sur la définition d'un vocabulaire de tokens. Lors de l'évaluation, la gestion des termes **Out-of-Vocabulary (OOV)** est un problème
- Si le corpus grandit, la dimensionnalité des vecteurs TF-IDF ou BOW grandit également

Prétraiter le langage

Pourquoi prétraiter ?

- **Limiter la dimensionnalité des vecteurs** (e.g. retrait des articles et pronoms qui portent peu de sens pour réduire le vocabulaire)
- Eviter de masquer la colinéarité des données en séparant sur plusieurs composantes des tokens sémantiquement proches (e.g. *manger* vs. *mangeais* vs. *mangerez* vs. *mnger* ...)

Familles de traitements

- **Traitements typographiques**
  - **Lowercasing** : uniformisation de la casse d'un document
  - **Normalisation** : remplacement/retrait de caractères spéciaux
  - **Correction orthographique**
- **Traitements morphologiques** (affectent la forme des mots et la lisibilité de la phrase)
  - **Lemmatisation** : un mot est remplacé par son lemme (e.g. *mangerai* $\rightarrow$ *manger* ; *aurions* $\rightarrow$ *avoir*)
    - La lemmatisation est une tâche complexe et coûteuse (e.g. *nous avions des avions*) qui doit s'appuyer sur la connaissance de la nature et la fonction du token
    - La complexité de la lemmatisation augmente avec la richesse de flexion (conjugaisons, accords...) de la langue
  - **Stemming (racinisation)** : application des règles pour tronquer les tokens à leur racine
  - **Retrait de stopwords** : retrait des mots indispensables à la structure de la phrase, mais ne portant pas de sens important (e.g. *pour*, *de*, *le*, *la*, *je*, *il*, *qui* ...)

Exemples d'applications

Pipeline de classification de texte[3]

1. Récupération du corpus de documents
2. Définition du type system (noms des classes) et annotation
3. Prétraitements
4. Vectorisation des **documents** (souvent par TF-IDF)
5. Entrainement du modèle (typiquement SVM, Logistic Regression...) à partir des vecteurs

[3] Ce cas d'usage sera traité en TP

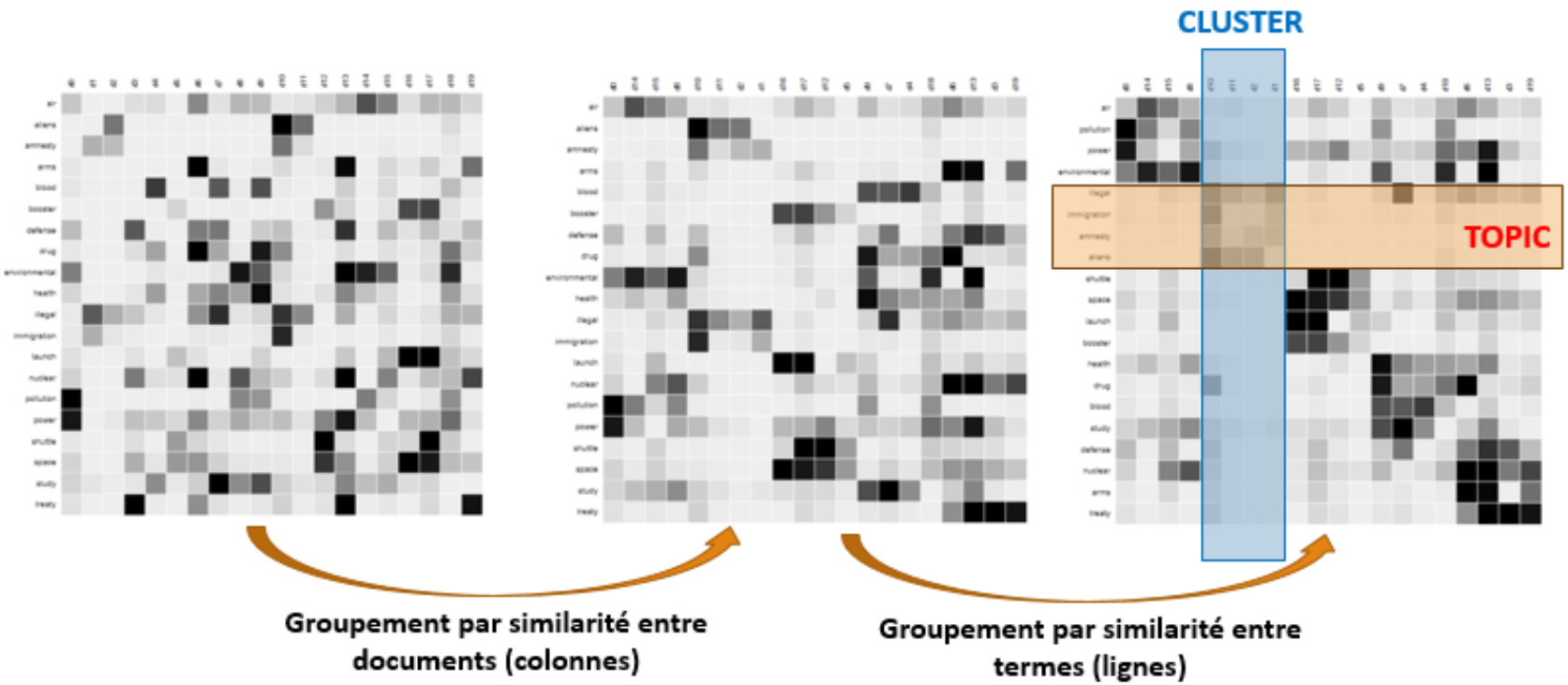
Pipeline d'extraction d'entités (NER)

1. Récupération du corpus de documents
2. Définition du type system (noms des entités) et annotation
3. Prétraitements
4. Vectorisation des **tokens**[4] du document.
5. Entrainement du modèle (typiquement CRF) à partir des vecteurs

[4] En séance 2, on verra des techniques de vectorisation mieux adaptées à ce cas d'usage

Pipeline de reconnaissance des topics : le Latent Semantic Indexing (LSA)

1. Récupération du corpus de documents
2. Prétraitements
3. Vectorisation des documents (typiquement par TF-IDF)
4. Clustering par similarité des vecteurs représentant les documents
5. Clustering par similarité des termes



Bibliographie

Bibliographie


Elman, J. L. (1990).
Finding structure in time.
Cognitive science, 14(2):179–211.


Jordan, M. (1986).
Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986.
Technical report, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science.


Koplenig, A. (2018).
Using the parameters of the zipf–mandelbrot law to measure diachronic lexical, syntactical and stylistic changes–a large-scale corpus analysis.
Corpus Linguistics and Linguistic Theory, 14(1):1–34.


Mandelbrot, B. (1965).
Information theory and psycholinguistics.
BB Wolman and E.