Représentation du lexique

Frederic Bechet, Carlos Ramisch, Benoît Favre
équipe TALEP, LIS UMR7020
Aix Marseille Université

Qu'est qu'un lexique ?

Il est question de vocabulaire !

- ***Lexique*** = Ensemble des lexies d'une langue, sans distinction flexionnelle
  - *Lexie* = Élément unitaire du lexique (lexèmes, locutions, proverbes)
  - *Lexème* = Représentation abstraite d'un mot englobant l'ensemble de ses réalisations

- ***Texte*** = Suite ordonnée de mots écrits.
  - *Mot* = Succession de sons dans les langues parlées, ou de signes dans les langues des signes ou écrites, qui a un sens propre, et qui est isolée soit par deux blancs à l'écrit, soit par une pause à l'oral

- Problématique
  - *Qu'utilise-t-on comme critère pour choisir les lexèmes et lexies pour une application de TAL ?*
  - *Comment représente-t-on les lexèmes et les lexies dans un ordinateur ?*
  - *Comment peut-on passer des mots aux lexies durant l'analyse d'un texte ?*

Comment sont codés les mots dans les machines ?

- 2 méthodes
  - Représentation explicite sous forme de dictionnaire
    - la liste des mots (lexies) connus est fixe, déterminée à l'avance
    - nécessité de prévoir un « plan B » pour les mots « hors vocabulaire » (Out-Of-Vocabulary – OOV words)
  - Représentation implicite apprise par rapport à un critère statistique
    - notion de « mots » dirigée par la tâche (transcription, traduction, ..)
    - pas de liste *a priori*, agglutination de phonèmes ou morphèmes à partir d'un corpus d'apprentissage

Représentation explicite des mots sous forme de dictionnaire

- Choisir une liste de mots (lexies)
  - Par rapport à des critères linguistiques ou à une application
- Clé d'accès de chaque entrée dans le dictionnaire
  - Forme orthographique (ex: « cour »)
  - Forme phonétique (ex: /kur/)
- Ajout d'informations pour chaque entrée
- Exemple : wiktionary
  - https://fr.wiktionary.org/wiki/cours

Clés d'accès pour les mots

- Forme orthographique ou phonétique
  - Ambigüités
    - « *les poules du **couvent** **couvent*** »
    - « je ***cours*** dans les ***cours*** de l'école après mes ***cours*** »
  - Modélisation des relations entre les mots
    - Morphologiques :
      - *cour/cours ; cheval/chevaux ; beau/belle ; suis/être*
    - Sémantiques : *animal => cheval ; cheval ⇳ jument ; cheval ⇳ poulain ..*
    - Autres
  - Exploitation des relations pour effectuer des « calculs » entre les mots

Associer des informations aux mots d'un dictionnaire

- Représenter la « nature » des mots
  - Formes (orthographique et/ou phonétique avec variantes)
  - Nature du mot (verbe, nom, adjectif, préposition, …)
    - **Part-Of-Speech : POS**
  - Lemme + morphologie
  - Informations fréquentielles (ex: TF/IDF)
- Codage sous forme informatique
  - Base de données structurée ou simple dictionnaire
  - Exemple : https://www.labri.fr/perso/clement/lefff/

Exemple d'entrées du Lefff

```
cours  100  cf   [pred="donner cours_____1<Suj:cln|sn,Objà:à-sn>",synt_head=donner]   donner cours_____1   Default      %default
cours  100  cfi  [pred="cours_____1<Suj:cln|scompl|sinf|sn>",lightverb=avoir]   cours_____1   Default      %default      inv
cours  100  cfi  [pred="cours_____2<Suj:cln|scompl|sinf|sn,Objà:à-sn>",lightverb=donner] cours_____2   Default      %default      inv
cours  100  nc   [pred="cour_____1<Objde:(de-sinf|de-sn),Objà:(à-sinf)>",cat=nc,@fp]   cour_____1   Default fp   %default      nc-2f
cours  100  nc   [pred="cours_____1<Objde:(de-sinf|de-sn),Objà:(à-sinf)>",cat=nc,semtype=event|-,@m]   cours_____1   Default m    %default      nc-1m
cours  100  v    [pred="courir_____1<Suj:cln|scompl|sinf|sn,Obj:(cla|sn)>",@imperative,@pers,cat=v,@Y2s] courir_____1   Imperative   Y2s   %actif  v-courir
cours  100  v    [pred="courir_____1<Suj:cln|scompl|sinf|sn,Obj:(cla|sn)>",@pers,cat=v,@P12s]   courir_____1   Default P12s   %actif  v-courir
cours  100  v    [pred="courir_____1<Suj:cln|sn,Obj:(cla|sn)>",@imperative,@pers,cat=v,@Y2s]   courir_____1   Imperative   Y2s   %actif  v-courir
cours  100  v    [pred="courir_____1<Suj:cln|sn,Obj:(cla|sn)>",@pers,cat=v,@P12s]   courir_____1   Default P12s   %actif  v-courir
cours  100  v    [pred="courir_____2<Suj:cln|sn,Objde:(de-sn|en),Obl:(sinf)>",@CtrlSujObl,@imperative,@pers,@être,cat=v,@Y2s]   courir_____2   Imperative   Y2s   %actif  v-courir
cours  100  v    [pred="courir_____2<Suj:cln|sn,Objde:(de-sn|en),Obl:(sinf)>",@CtrlSujObl,@pers,@être,cat=v,@P12s]   courir_____2   Default P12s   %actif  v-courir
cours  100  v    [pred="courir_____2<Suj:cln|sn>",@imperative,@pers,cat=v,@Y2s]   courir_____2   Imperative   Y2s   %actif  v-courir
cours  100  v    [pred="courir_____2<Suj:cln|sn>",@pers,cat=v,@P12s]   courir_____2   Default P12s   %actif  v-courir
cours  100  v    [pred="courir_____3<Suj:cln|sn,Obl:après-sn>",@imperative,@pers,cat=v,@Y2s]   courir_____3   Imperative   Y2s   %actif  v-courir
cours  100  v    [pred="courir_____3<Suj:cln|sn,Obl:après-sn>",@pers,cat=v,@P12s]   courir_____3   Default P12s   %actif  v-courir
cours  100  v    [pred="courir_____4<Suj:cln|sn,Loc:loc-sn>",@imperative,@pers,cat=v,@Y2s]   courir_____4   Imperative   Y2s   %actif  v-courir
cours  100  v    [pred="courir_____4<Suj:cln|sn,Loc:loc-sn>",@pers,cat=v,@P12s]   courir_____4   Default P12s   %actif  v-courir
```

Exemple de format simplifié : indice + mot + freq + liste de couple pos, lemme

```
094571 cours   5233   nc     cour    nc     cours   v      courir
081027 comme_ça      5231   adv    comme_ça      csu    comme_ça
419837 système 5209   nc     système
065207 celle   5184   pro    celui
305576 pas_de  5182   det    pas_de
256284 le_plus 5182   adv    le_plus
214163 gouvernement   5175   nc     gouvernement
066100 certains      5154   adj    certain det    certain nc     certain pro    certains
412212 sud     5153   adj    sud    nc     sud
376985 rien    5117   nc     rien   pro    rien
324421 pouvoir 5107   nc     pouvoir v      pouvoir
278845 mm      5097   nc     millimètre
065782 ce_qu'  5097   csu    ce_que
104764 début   5085   nc     début
441935 vais    5074   v      aller
200300 forme   5065   nc     forme   v      former
```

Associer des informations aux mots d'un dictionnaire

- Représenter le « sens » des mots
  - Représentations symboliques explicites
    - Lexique sémantique avec relations (ex: WordNet)
  - Représentations continus (analyse statistique de corpus)
    - Représentation vectorielle
    - Modèles de Langage

Représentation symbolique du « sens » d'un mot : WordNet

Représentations symboliques

- synsets

Liste prédéfinie de relations

- synonymes
- antonymie
- hyperonymes
- méronymes
- ….

Avantages

- Précision linguistique
- Représentation informatique

Inconvénients

- Manque de couverture
- Ressource « statique » => pas d'adaptation au contexte

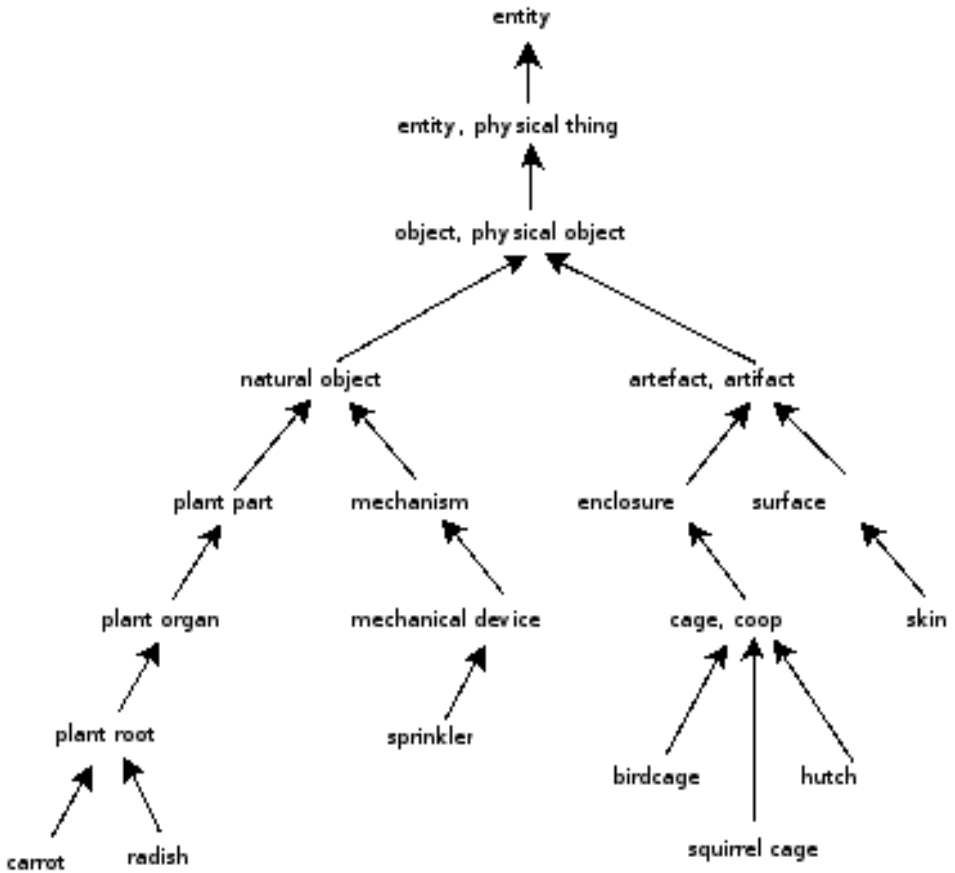


Figure 1. "is a" relation example

Représentation continue du « sens » d'un mot

- Sémantique distributionnelle

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

  - Une des idées majeures du Traitement Automatique de la Langue
  - De très nombreux modèles statistiques de calculs de similarités
  - Notion de contexte : global (document) ou local (phrase)
  - Représentation vectorielle des mots
  - Extraction de relations entre vecteurs

Représentation continue du « sens » d'un mot

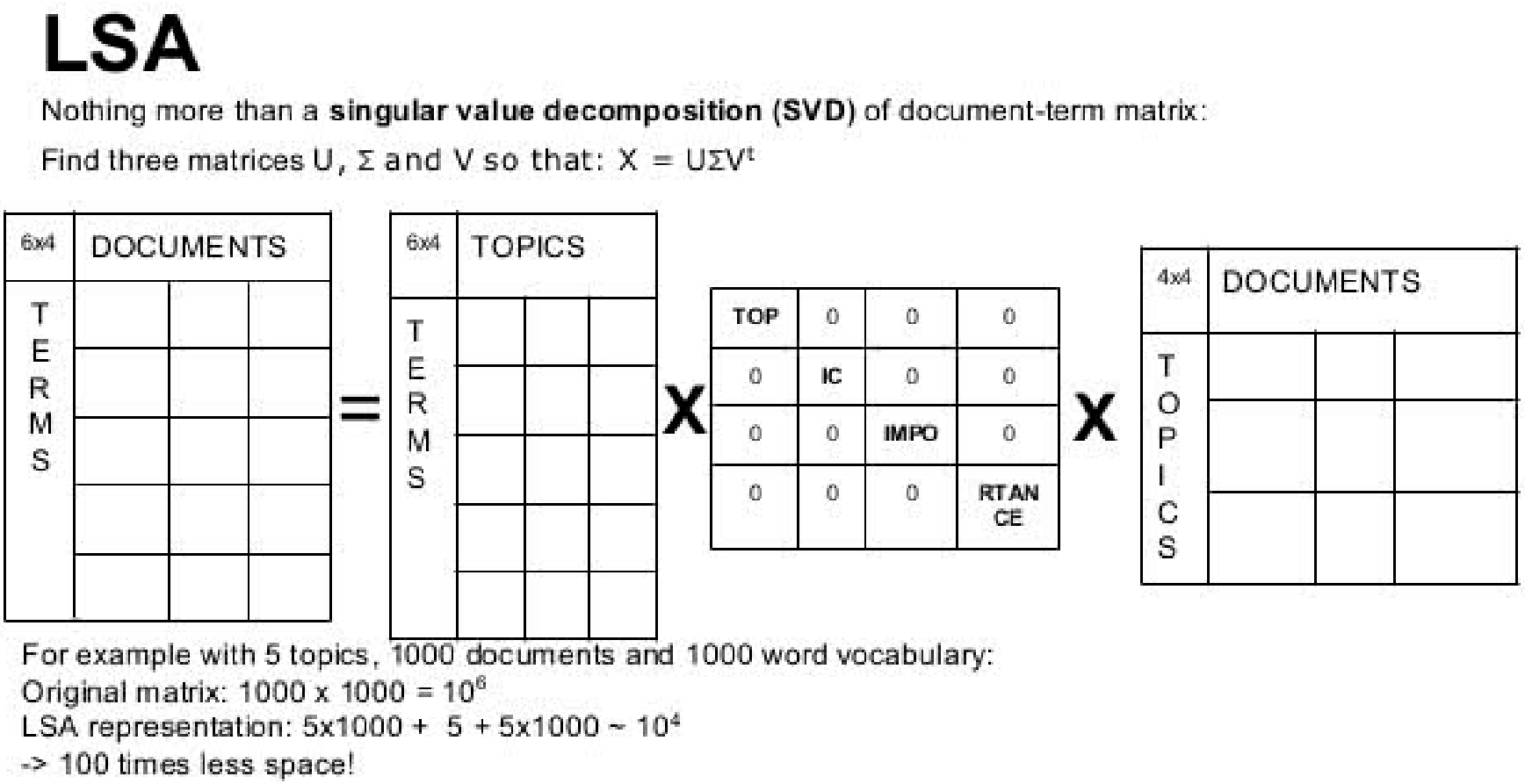
- *words which are similar in meaning occur in similar contexts* (Rubenstein & Goodenough, 1965)
- *. . . In other words, difference of meaning correlates with difference of distribution* (Harris, 1970, p.786)
- *words with similar meanings will occur with similar neighbors if enough text material is avai*lable (Schutze & Pedersen, 1995)
- *a representation that captures much of how words are used in natural context will capture much of what we mean by meanin*g (Landauer & Dumais, 1997)
- *in the proposed model, it will so generalize because "similar" words are expected to have a similar feature vector, and because the probability function is a smooth function of these feature values, a small change in the features will induce a small change in the probability.* (Bengio et al, 2003)

Représentation continue du « sens » d'un mot

- Modèles « fréquentiels »
  - une matrice de "comptes" de co-occurrences
    - mots / documents ou mots / contexte
  - schéma de pondération des comptes (PMI, LLR, etc.)
  - principe de réduction de dimensionnalité de la matrice
    - Singular Value Decomposition [Golub and Van Loan, 1996]
    - non-negative matrix factorization [Lee and Seung, 1999]
    - ..
  - Chaque ligne dans la matrice réduite représente un vecteur de mots
  - Calcul de distances entre vecteurs pour estimer des similarités
    - Ex: similarité cosine

Représentation continue du « sens » d'un mot

- Exemple : Latent Semantic Analysis



Représentation continue du « sens » d'un mot

- Les modèles de langage
  - Comment estimer la « probabilité » d'un mot ou d'unse séquence de mots ?
- Application
  - Pour toutes les applications nécessitant de « générer » du langage
    - Traduction, transcription parole, résumé, dialogue, etc.
  - Comment choisir parmi toutes les phrases « possibles » ?

Représentation continue du « sens » d'un mot

- Choix des mots et ordre des mots
- On aimerait que le score $f(s_k)$ trie les phrases ci-dessous :
  - (+++) *le chat boit du lait*
  - (++) *le chameau boit du lait*
  - (+) *la chaise boit du lait*
  - (−) *chat le boit lait du*
  - (−−) *chat boit lait*
  - (−−−) *bai toht aict*
- Si $w_1 \dots w_m$ est une séquence de mots, comment calculer $P(w_1 \dots w_m)$?
- ***Idée*** : estimer $P(w_1 \dots w_m)$ à partir d'un échantillon de la langue, appelé **corpus d'entraînement**

Représentation continue du « sens » d'un mot

Probabilité d'une phrase ?

- Estimation avec un gros ensemble de textes

$$P(w_1 \dots w_m) = \frac{nb(w_1 \dots w_m)}{nb(\text{phrases possibles})}$$

- Avantages :
  - modèle très précis
  - probabilités simples à calculer
- Inconvénients :
  - estimations nécessitent un échantillon trop gros
  - la plupart des phrases n'apparait jamais dans un corpus
  - la plupart des phrases aura une probabilité estimée de zéro

Représentation continue du « sens » d'un mot

- Solution possible :
  - les mots d'un texte sont le résultat d'un processus statistique modélisable sous la forme d'une « chaîne de Markov »
  - processus Markovien

Exprimer la séquence en fonction de ses sous-parties

$$
\begin{aligned}
P(w_1 \dots w_m) &= P(w_m | w_{m-1} \dots w_1) P(w_{m-1} \dots w_1) \\
&= P(w_m | w_{m-1} \dots w_1) P(w_{m-1} | w_{m-2} \dots w_1) \\
&= P(w_1) \prod_i P(w_i | w_{i-1} \dots w_1)
\end{aligned}
$$

Note : on ajoute deux symboles $\langle s \rangle$, $\langle /s \rangle$ en début et fin de chaîne.

Représentation continue du « sens » d'un mot

- Exemple

$$
\begin{aligned}
P(\langle s\rangle \text{le chat boit du lait} \langle /s\rangle) = & P(\langle s\rangle) \\
& \times P(\text{le}|\langle s\rangle) \\
& \times P(\text{chat}|\text{le}) \\
& \times P(\text{boit}|\text{le chat}) \\
& \times P(\text{du}|\text{le chat boit}) \\
& \times P(\text{lait}|\text{le chat boit du}) \\
& \times P(\langle /s\rangle|\text{le chat boit du laitt})
\end{aligned}
$$

Représentation continue du « sens » d'un mot

Chaîne de Markov (Modèle n-gramme)

- Approximation : hypothèse "d'horizon k" (hypothèse de Markov) :

$$P(mot(i)|historique(1, i-1)) \simeq P(mot|historique(i-k, i-1)$$
$$P(w_i|w_1 \dots w_{i-1}) \simeq P(w_i|w_{i-k} \dots w_{i-1})$$

- Pour $k = 2$

$$\begin{aligned} P(\langle s\rangle \text{le chat boit du lait} \langle /s\rangle) \simeq & P(\langle s\rangle) \times P(\text{le}|\langle s\rangle) \times P(\text{chat}|\text{le}) \\ & \times P(\text{boit}|\text{le chat}) \times P(\text{du}|\text{chat boit}) \\ & \times P(\text{lait}|\text{boit du}) \times P(\langle /s\rangle|\text{du lait}) \end{aligned}$$

- Estimation par maximum de vraisemblance

$$P(\text{boit}|\text{le chat}) = \frac{nb(\text{le chat boit})}{nb(\text{chat boit})}$$

- Modèle n-gramme ($n = k + 1$), utilise n mots pour l'estimation
  - $n = 1$: unigramme, $n = 2$: bigramme, $n = 3$: trigramme...

Représentation continue du « sens » d'un mot

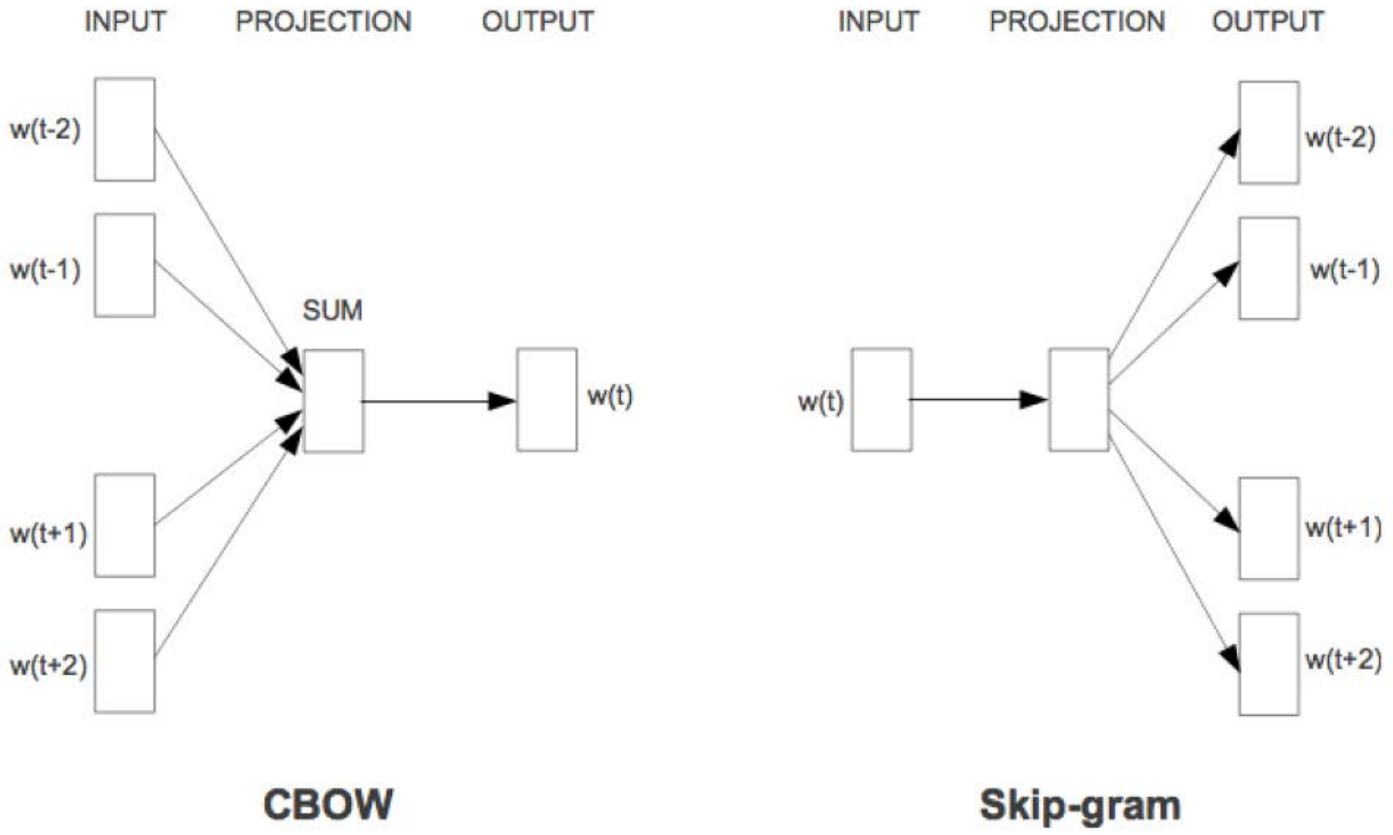
- Modèles de langage
  - Evaluation ?
    - Mesure de la Perplexité (capacité du modèle à prédire un texte)
  - Modèles de Markov
    - Avantages : simplicité / efficacité pour les évènements vus
    - Limitations : problème des évènements non vus
      - N-grammes absents ou mots hors vocabulaire (Out Of Vocabulary word)
      - Limitation du contexte pris en compte : n-grammes avec 1<n<5
  - Modèles de langage neuronaux
    - Prédiction du « mot d'après » par un réseau de neurone
    - Avantages :
      - meilleure généralisation aux évènements non vus
      - augmentation possible des contextes de prédiction
    - Inconvénients
      - Beaucoup plus « gourmands » en terme de ressources (corpus et capacité de calcul) pour les entraîner

Représentation continue du « sens » d'un mot

- Modèles « prédictifs » (ou probabiliste)
  - Apprendre sur un (gros) corpus de texte un modèle pour :
    - Prédire un mot à partir de son contexte d'occurrence
    - Prédire un contexte d'occurrence à partir d'un mot
  - Modèle d'apprentissage basé sur les réseaux de neurones
    - Pour chaque prédiction liée à un mot, on récupère une couche de paramètres dans le réseau
    - Cette couche est le vecteur représentant le mot !!
  - « sucess story » autour des « **word embeddings** »
    - **Mikolov 2013 : Word2Vec**
    - « *Don't count, predict !* » [Baroni et al., 2014]

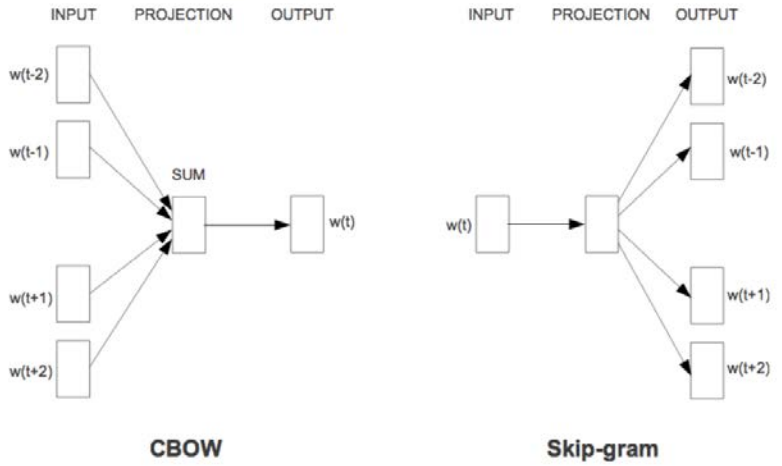
Représentation continue du « sens » d'un mot

- Distributed representations of words and phrases and their compositionality
  - T **Mikolov**, I Sutskever, K Chen, GS Corrado… - NIPS 2013 (Cité 9707 fois)



Représentation continue du « sens » d'un mot

- Distributed representations of words and phrases and their compositionality
  - T **Mikolov**, I Sutskever, K Chen, GS Corrado… - NIPS 2013 (Cité 9707 fois)



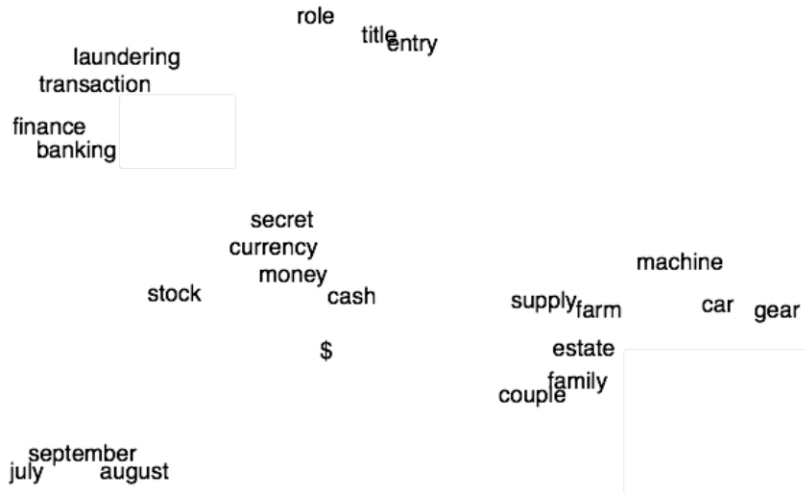
  - Arithmétique analogique des représentations
    - vec(*Madrid*) - vec(*Spain*) $\simeq$ vec(*Paris*) - vec(*France*)
    - permet de résoudre des équations analogiques : [x: y :: z :?]

1. calculer $t = vec(y) - vec(x) + vec(z)$ le vecteur cible
2. rechercher dans V, le mot $\hat{t}$ le plus proche de t :

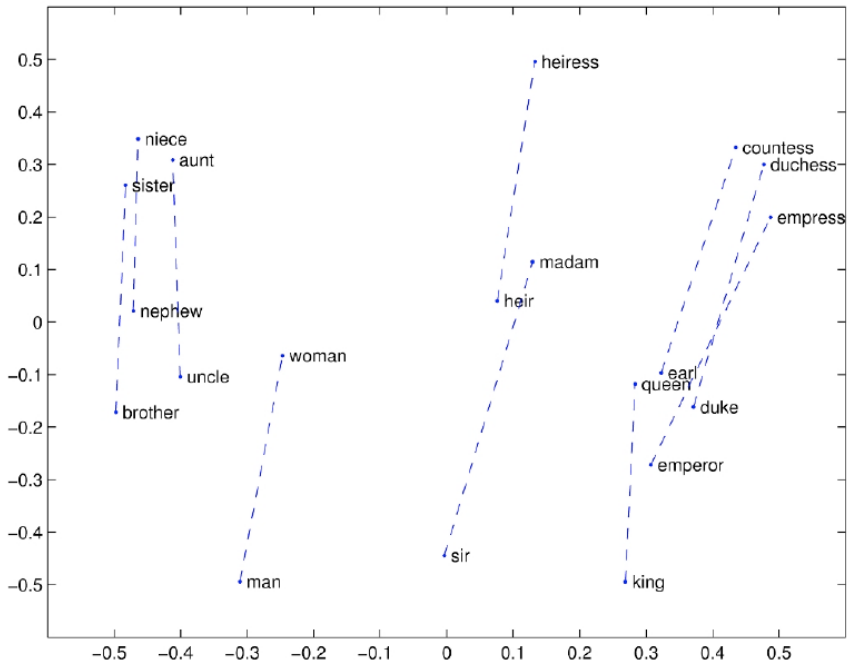
$$\hat{t} = \underset{w}{\operatorname{argmax}} \frac{vec(w).vec(t)}{||vec(w)|| \times ||vec(t)||}$$

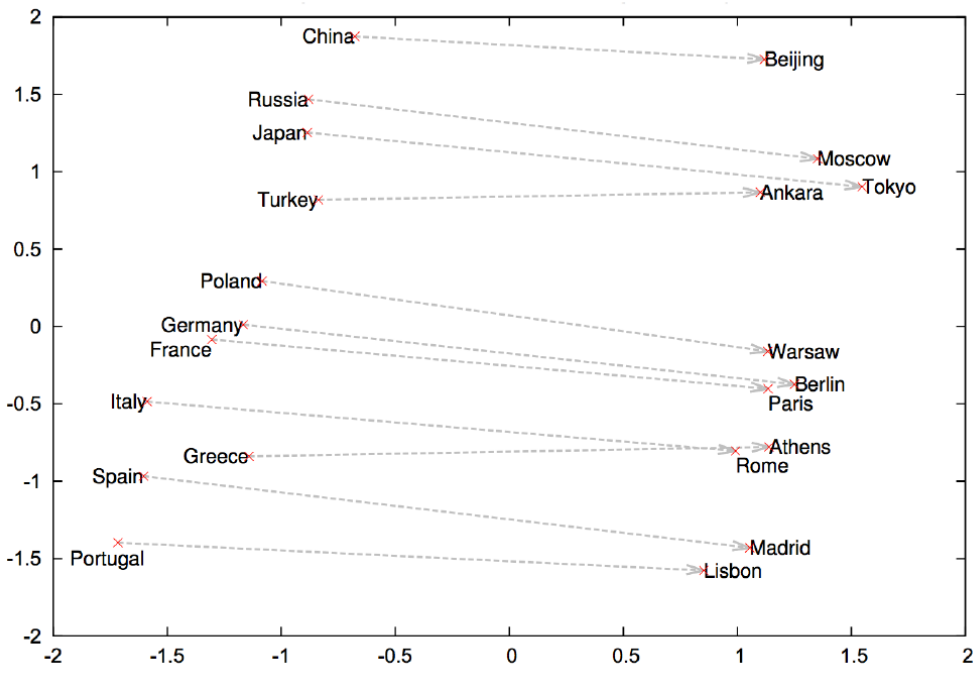
Exemples

$$currency = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$



Demo:
https://pageperso.lis-lab.fr/benoit.favre/tsne-frwiki/





Comment sont codés les mots dans les machines ?

- Représentation implicite apprise par rapport à un critère statistique
  - Représentations apprises avec des modèles prédictifs au niveau des successions de symboles élémentaires
    - Caractères (lettres, ponctuations, signes, ..)
    - Phonèmes
  - Choix des unités qui maximisent la « qualité » de la prédiction
    - Critère de la Perplexité (cross-entropie)
  - Représentations des unités sous forme de vecteurs
    - Correspondant à un « couche » de paramètres d'un réseau de neurones
  - Démo : http://talepdemo.lis-lab.fr/index-gutemberg.html

Tendance actuelle

- Dynamic contextual representation (Transformer models)
  - Chaque occurrence de mot à un embedding différent
    - Calculé « à la volée » en traitant chaque phrase
    - Appris sur des quantités gigantesque de texte
    - Et sur des tâches ne nécessitant aucune supervision humaine (ex: masquage)
  - Avantage
    - Plus de problème de polysémie des embeddings unique
    - Mélange lexique/modèle de langage
    - Gain important en performance sur la plupart des tâches de TAL
- Vocabulaire ouvert
  - Utilisation de « mots » pour les token les plus courants (et les plus courts)
  - Décomposition des autres mots en « word piece », unités sous-lexicales
    - Exemple : « I like strawberries » => « I like straw ##berries »
  - Plus de mots hors-vocabulaire !!
  - …. mais plus de dictionnaire explicite !!

Tendance actuelle

- Example:
  - **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**

    Jacob Devlin Ming-Wei Chang, Kenton Lee, Kristina Toutanova , 2018
    - Pros: very efficient !! big increase in performance for many tasks
    - Cons: nearly impossible to learn from scratch if you are not a GAFAM !!

  - GPT-3
    - GPT-3 est un modèle de langage développé par la société OpenAI annoncé le 28 mai 2020 et ouvert aux utilisateurs via l'API d'OpenAI en juillet 2020.
    - Au moment de son annonce, GPT-3 est le plus gros modèle de langage jamais entraîné avec 175 milliards de paramètres. GPT-2, sorti en 2019, n'avait que 1,5 milliards de paramètres2.

De nouvelles préoccupations

Energy and Policy Considerations for Deep Learning in NLP
Emma Strubell, Ananya Ganesh, Andrew McCallum
ACL 2019

Consumption	CO_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Découpage d’un texte en unité de base

TOKENISATION

Comment passer d’un texte, flux de caractères, composé de « mots »
vers une séquence de « lexies » parfaitement identifiées ?

Tokenisation

Lexique

- Un texte est une séquence de caractères
- En TAL, il est intéressant de voir un texte comme une séquence de mots
- Le **lexique** représente l'ensemble des unités lexicales (mots) du texte
  - Aussi appelé dictionnaire ou vocabulaire
- Un lexique peut être donné à priori (notre cas) ou inféré à partir du texte
- Comment représenter les unités lexicales (mots) dans un modèle de TAL ?

Tokenisation

Lexique et internalisation de chaînes de caractères

- Chaînes de caractères - opérations lentes
  - Comparaisons, tri, etc.
- Idée : convertir chaque mot en un entier
  - Un texte peut être représenté avec `N * sizeof(int)` octets
  - Le lexique donne la correspondance entiers-mots
- Comment représenter le lexique ?

Tokenisation

Lexiques

- Correspondance mots $\leftrightarrow$ codes entiers
  - D'autres informations telles que les POS peuvent y figurer
- Rechercher mot de n caractères parmi N mots
- Structures de données - recherche d'une valeur :
  - Tableau associatif trié - $O(logN)$ en moyenne
  - Tables de hash - $O(n)$ au mieux, $O(N)$ au pire
  - Arbres de préfixes/en parties communes - $O(n)$ (selon l'implémentation)

Tokenisation

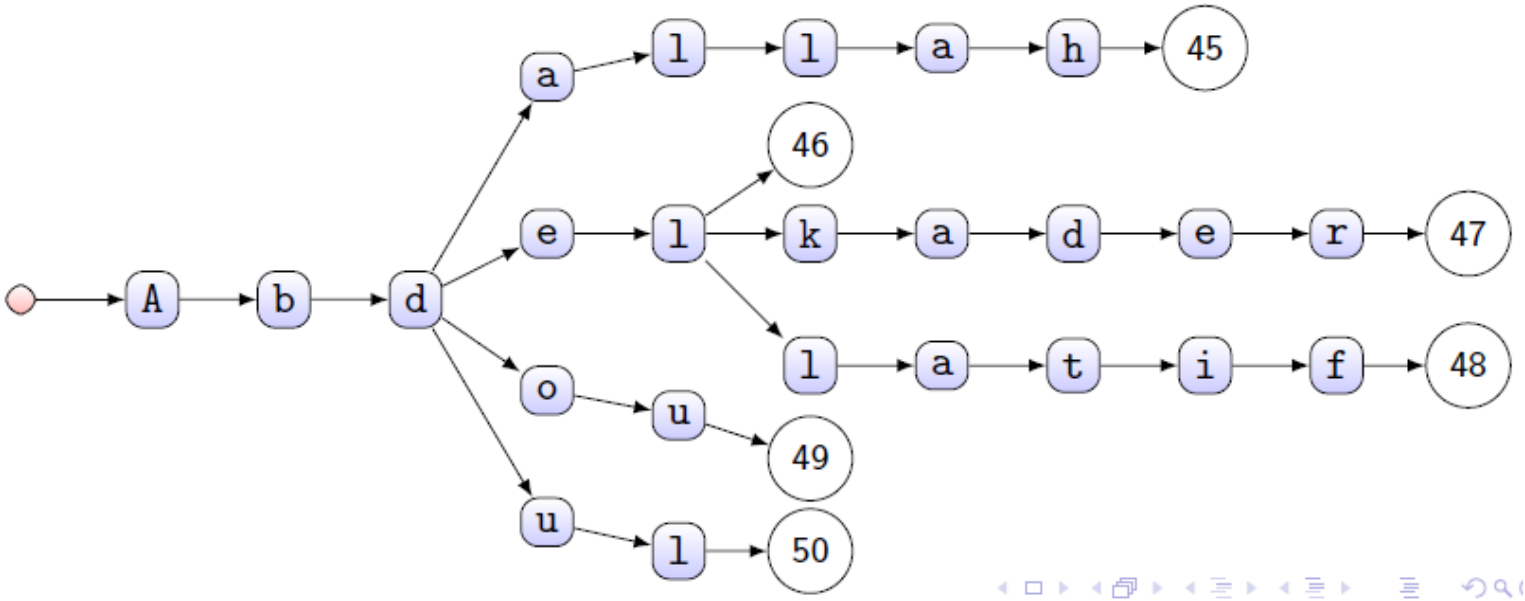
Arbre de préfixes/en parties communes

- Structure de données pour représenter le lexique de manière efficace
- Chaque chemin de la racine jusqu'à une feuille correspond à un mot
- Chaque noeud correspond à un caractère du mot
- Les préfixes communs sont factorisés
- On sépare les mots à partir du moment où ils diffèrent

Tokenisation

Exemple

```
...
45 Abdallah
46 Abdel
47 Abdelkader
48 Abdellatif
49 Abdou
50 Abdul
```



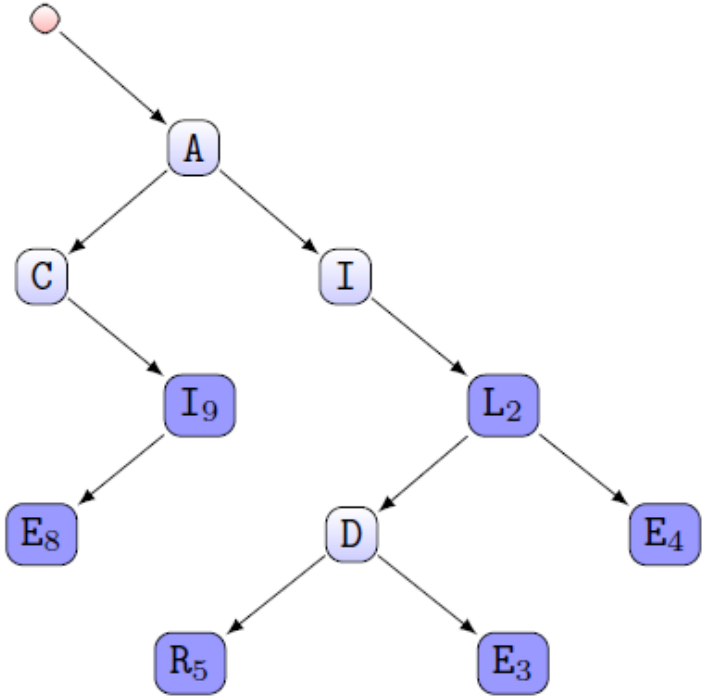
Tokenisation

Implémentation

- Il s'agit d'un arbre **n-aire**, avec un nombre variable de fils par noeud
- *Alternative 1* : chaque noeud contient une liste de fils
  - Un tableau classique, si l'alphabet fixe et petit
  - Une liste chaînée (triée)
  - Une table de hachage
- *Alternative 2* : représenter un arbre n-aire par un arbre binaire
  - Chaque noeud contient un caractère, un fils gauche et un fils droit
  - Les fils droits contiennent le premier fils du noeud
  - Les fils gauches contiennent les fils suivants (frères)
- Le code est un champ du noeud – il vaut -1 quand il n'y a pas de code

Tokenisation

Exemple



- Quels mots sont représentés dans ce lexique ?
  - AIL (2), AIDE (3), AILE (4), AIR (5), CE (8) et CE (9)

Noeuds bleu foncé = contiennent un code en indice différent de -1

Tokenisation

Création d'un arbre de préfixes

- Pour chaque nouveau mot w de longueur n (+ son code c), il faut :
  1. Lire le mot w caractère par caractère $w[i], i = 1..n$
  2. Suivre le chemin correspondant dans l'arbre
  3. Si on arrive à un noeud avec le code c :
     - ⋆ Rien à faire (mot est déjà présent)
  4. Si on arrive à un noeud sans code :
     - ⋆ Ajouter le code c au noeud (préfixe d'un mot existant)
  5. Si on est bloqué (aucun caractère ne correspond à $w[i]$)
     - ⋆ Ajouter une branche pour le suffixe $w[i : n]$ avec c à la feuille

Tokénisation

- Rechercher les unités du lexique dans le texte
- Comment prendre en compte les segmentations ambiguës ?
- *La durée* ou *Ladurée* ?
  - Apprentissage : modèles à n-grammes
  - Heuristiques : **match le plus long**
- Problème : out-of-vocabulary (OOV)
  - Problème récurrent en TAL
  - Loi de Zipf
- Éléments spéciaux : dates, URLs, etc
  - Expressions régulières

Tokénisation avec arbre de préfixes

- Parcourir l'arbre et le texte en parallèle
- Quand on arrive à un séparateur (p.ex. un blanc)
  - Si on a un code ($\neq -1$) dans le noeud de l'arbre :
    - on sauvegarde le code c
    - on sauvegarde la position dans le texte i
    - on continue
  - Sinon, on continue
- Quand on arrive à un noeud sans suite :
  - Si le prochain symbole est un séparateur et on a un code :
    - on renvoie le code du noeud actuel
  - Sinon, si le prochain symbole est un séparateur :
    - on renvoie le dernier code c sauvegardé
    - on revient à la position i
  - Sinon
    - on renvoie 0 (mot inconnu)
    - on avance jusqu'au prochain séparateur
  - On recommence à partir de la racide de l'arbre

Tokenisation : Reconnaissance et traduction des tokens

Texte "brut"

L'Australie achève samedi par un match à hauts risques en Argentine son décevant Four Nations, qui a suscité une vive inquiétude au pays et placé l'entraîneur Robbie Deans en position précaire. Les Wallabies, vainqueurs du Tri-Nations l'an dernier et troisièmes de la Coupe du monde dans la foulée, pointent en avant-dernière place (2 victoires, 3 défaites) juste devant les novices argentins. L'ultime rencontre samedi (20h10 locales, 01h10 en France) à Rosario a tout du traquenard face à des Pumas déterminés à remporter devant un public toujours chaud une première victoire dans le tournoi, après avoir tenu en échec les Springboks (16-16). Avec un groupe encore incertain (Barnes, Ashley-Cooper, Ioane, Samo, Polota Nau touchés samedi dernier) et en prévision d'un rude combat, l'encadrement devrait muscler considérablement son pack.

Texte "tokenisé"

```
<s> l' Australie achève samedi par un match à hauts risques en Argentine son décevant
Four Nations , qui a suscité une vive inquiétude au pays et placé l' entraîneur Robbie
Deans en position précaire . </s>
<s> les Wallabies , vainqueurs du Tri - Nations l' an dernier et troisièmes de la
Coupe du monde dans la foulée , pointent en avant - dernière place ( deux victoires ,
trois défaites ) juste devant les novices argentins . </s>
<s> l' ultime rencontre samedi ( 20h10 locales , 01h10 en France ) à Rosario a tout
du traquenard face à des Pumas déterminés à remporter devant un public toujours chaud
une première victoire dans le tournoi , après avoir tenu en échec les Springboks (
seize moins seize ) . </s>
<s> avec un groupe encore incertain ( Barnes , Ashley - Cooper , Ioane , Samo , Polota
Nau touchés samedi dernier ) et en prévision d' un rude combat , l' encadrement
devrait muscler considérablement son pack . </s>
```

Texte « traduit » en séquences de symboles

1054 7815 4238 9297 6283
5831 1054 4554 688 9296 7960 8104 9297 757 8908 1203 6861 2624 4691 7802 4772 18 789 7815 4238 9297 6283 4691 7802 4771 4772 18 789 3311 9297 5964 2279 5818 5909