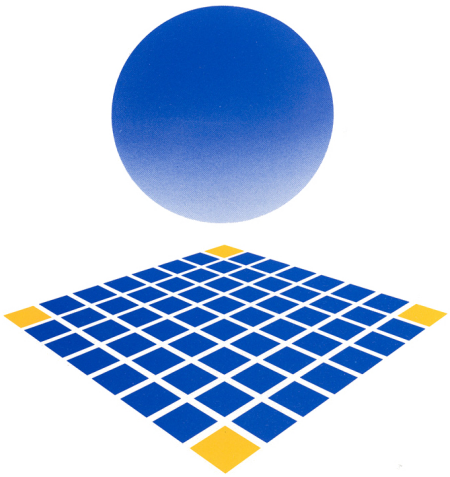

Université Montpellier II
Faculté des Sciences

Traitement Automatique du Langage Naturel

Sémantique et fouille de textes

TOTAKI

Auteur :
Kevin Cousot
Rider Carrion

Enseignants :
Mathieu Lafourcade

6 janvier 2014

Résumé

TODO 1.

Mots-clés

Traitement Automatique du Langage Naturel, réseaux lexico-sémantique, analyse sémantique.

Table des matières

CHAPITRE 1

Introduction

TODO 2. *Quelques mots sur le contexte et la problématique*

1.1 Jeux de mots

1.1.1 Présentation

Jeux de mots [1] (JDM) est un jeu en ligne ou les parties des joueurs sont utilisés pour construire un réseau lexical.

Le déroulement d'une partie est simple. D'abord une consigne (exemple : « Donner des idées associés à ») et un mot (exemple : « parasol ») s'affichent. Le joueur a alors une minute pour proposer ses réponses. Elles sont ensuite comparées à celles d'un joueur ayant joué cette partie plus tôt et les mots communs leur rapportent des points. D'autres mécanismes, comme les « duels » ou les « procès » incitent les joueurs à jouer plus et ainsi étendre et consolider le réseau.

Un des avantages de l'approche est que le réseau lexical est évolutif, les parties des joueurs permettent de constantes mises à jour. De plus, le système étant présenté sous la forme d'un jeu les résultats obtenus sont plus naturels et représentatifs que ceux fournis pas des experts (bien que moins précis).

Au moment de la rédaction de ce document, le réseau de JDM possède 289 873 termes et 4 967 486 relations.

1.1.2 Réseau lexical

Le TALN a besoin de modèles pour représenter l'information lexical et sémantique, les réseaux lexicaux sont une possibilité. Un réseau lexical est un graphe dont les sommets sont des termes (mots ou expressions) et les arcs sont des relations binaires portant sur le lexique ou la sémantique des sommets. Ces relations sont pondérées afin d'en traduire l'importance. [1].

Dans le réseau lexical de JDM [2], on trouve plusieurs catégories de relation :
- relations lexicales : synonymie, antonymie ...
- relations ontologiques : générique (hyperonymie), spécifique (hyponymie), partie (méronymie), tout (holonymie).
- relations associatives : association libre, sentiment associé, signification.
- relations prédicatives : agent (sujet), patient (objet), instrument ...
- relations de typicalités : lieux, moments, caractéristiques typiques

TODO 3. *Donner un meilleur exemple, si possible issu de JDM*

On représente un réseau lexical sous la forme d'un graphe orienté :

$G = (V, E)$ avec
- V : l'ensemble des n sommets (mots ou expressions)
- $E \subseteq V \times V$: l'ensemble des m arcs (relations lexicales ou sémantiques)

1. http ://jeuxdemots.org

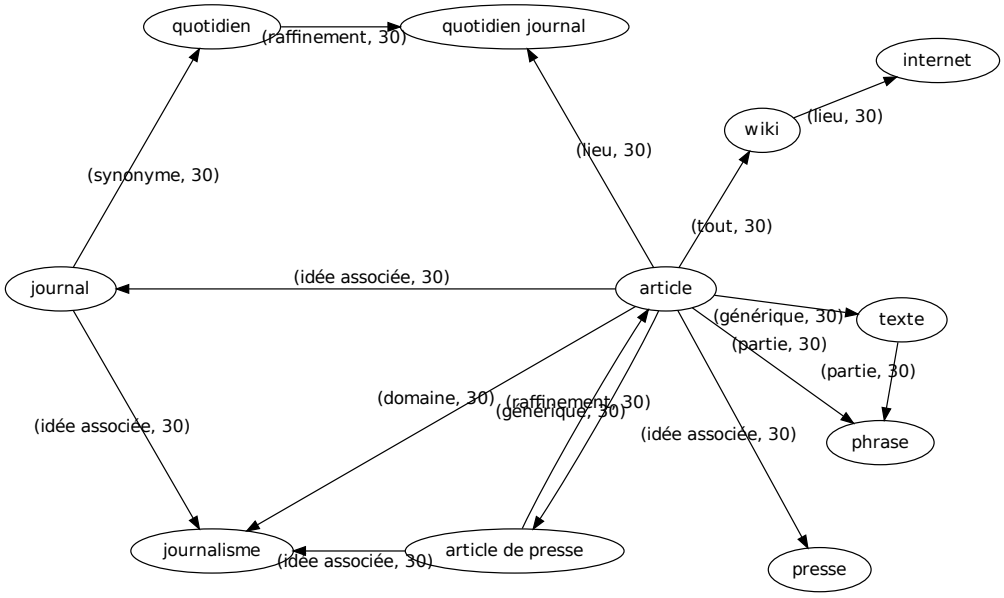

FIGURE 1.1 – Exemple de réseau lexical

1.1.3 Signature

Une signature est un ensemble fini et typé de termes pondérés donnant une définition par extension d'un terme. Les types sont ceux des relations du réseau et les poids sont la somme des poids des relations adjacentes.

Par exemple, la signature du sommet « article » en 1.1.2 est :

$$S(article) = (\text{raffinement} : 34, \text{idée} : 84)$$

TODO 4.
Exemple ***réel***
Qu'est-ce que ca représente ?
Notion définissable pour terme, concept, texte entier ...
Différence avec les vecteurs d'idées ?

1.2 TOTAKI

1.2.1 Présentation

TOTAKI[2] (Tip of the Tongue and Automated Knowledge Inferer) est également un jeu en ligne, il a pour objectif d'évaluer le réseau lexical de JDM [3]. Le principe est de faire deviner un mot au programme. Le joueur donne successivement des indices auxquels TOTAKI répond par une proposition. La partie s'arrête quand le mot est deviné ou si TOTAKI abandonne et demande la réponse. Il est possible de préciser la relation que le terme cherché a avec l'indice donné.

Quelques exemples de parties

Exemple 1 :
- méthode ⇒ technique
- informatique ⇒ algorithme ✓

Exemple 2 :
- `:loc` lampe ⇒ ampoule
- `:couleur` bleu ⇒ fil
- `:cause` frotter ⇒ génie ✓

Exemple 3 :
- logiciel ⇒ ordinateur
- messagerie ⇒ MSN
- canaux ⇒ client de messagerie
- protocole ⇒ informatique
- chat ⇒ ⊠

Réponse : IRC

2. http ://www.jeuxdemots.org/AKI2.php

1.2.2 L'algorithme

Une fois le premier indice i_1 donné, on commence par calculer sa signature $S(i_1)$ dont les termes sont triés par activations décroissantes. Le terme le plus activé, noté $max(S)$, est proposé comme réponse R_1 après avoir retiré l'indice de la signature. Il est lui même retiré à son tour pour obtenir la signature courante S'_1 :

$$\begin{aligned} S(i_1) &= S_1 = t_1, t_2, \dots \\ S'_1 &= S_1 \setminus i_1 \setminus t_1 \\ R_1 &= max(S_n) = t_1 \end{aligned} \tag{1.1}$$

Pour tout autre indice $i_{n>1}$, on fait l'intersection entre la signature courante et celle de i_n :

$$\begin{aligned} S_n &= (S'_{n-1} \cap S(i_n)) \setminus i_n \\ S'_n &= S_n \setminus max(S_n) \\ R_n &= max(S_n) \end{aligned} \tag{1.2}$$

A chaque itération, le nombre de termes de la signature courante diminue et termine par être vide si aucune réponse n'est correcte. Ce stade atteint, le système peut demander la réponse à l'utilisateur ou tenter un rattrapage. La procédure de rattrapage ne se base plus sur l'intersection de signatures, mais sur leur somme :

$$\begin{aligned} S_n &= (S'_{n+1} \oplus S(i_n)) \setminus i_t \\ S'_n &= S_n \setminus max(S_n) \\ R_n &= max(S_n) \end{aligned} \tag{1.3}$$

TODO 5.
Décrire la différence intersection / somme
Préciser que le nombre d'application de la procédure doit être limitée

1.3 Problématique

TODO 6.

CHAPITRE 2

Meilleur connecteur

Regardons l'algorithme donné 1.2.2 d'un point du vue graphe. Au premier indice (1.1), on répond par le terme le plus activé, c'est à dire le voisin le plus activé. A partir du second indice (1.2), l'intersection des deux termes est faite, et le terme le plus activé est proposé. Cela revient à obtenir les voisins directs communs aux termes, et proposer le plus activé.

On suivra ici la même logique, mais plutôt que de rechercher les voisins communs à distance 1, on cherchera les voisins communs à distance L avant de proposer le meilleur.

2.1 Calcul des chemins

Étant donné un ensemble d'indices $I \subseteq V$ de $k > 1$, on cherche à trouver l'ensemble des meilleurs connecteurs, noté $C_I \subseteq \bar{I}$, c'est à dire les sommets connectant le plus de paires d'indices. On appelle connecteur un sommet se trouvant sur au moins au chemin élémentaire reliant une paire d'indice. Plus le chemin est long, moins il est pertinent, on décide donc de limiter la longueur [1] à L.

Plus formellement, on se donne la fonction M_L qui, à une paire de sommets distincts, associe l'ensemble des chemins de longueur maximum L les connectant. On veut trouver les c :

$$C_I = \{c \in \bar{I} \text{ tel que } |\{(i,i') : (i,i') \in \mathcal{P}_2(I) / \exists \mu \in M_L(i,i') \text{ avec } c \text{ une extrémité dans } \mu\}|_{MAX}\}$$

Prenons par exemple le graphe de la figure 2.1. Les rectangles sont des indices ($I = \{A, B, C\}$) et les cercles des sommets quelconques. La direction des relations n'ayant pas d'importance, le graphe est non-orienté. Ici, avec $L = 3$, on a $C_I = \{1, 2, 3\}$ car ils connectent toutes les paires : A–B, A–C, B–C. Les sommets 4 et 5 ne connectent pas B–C et A–B respectivement.

Le calcul des chemins pour une paire d'indices (i, i') est possible à l'aide d'un parcours en profondeur.

TODO 7. *Expliquer algorithme*

TODO 8. *Complexité : $O(k.m)$.*

2.2 Recherche du meilleur connecteur

$$C_B = \sum_{\substack{\forall \mu \in M_L(i,i') \\ \forall (i,i') \in \mathcal{P}_2(I)}} i$$

[4] [5]

2.3 Implémentation, tests

2.4 Discussion

TODO 9. *Prendre en compte :*

1. On pourrait aussi fixer à un poids moyen (comme approximation de la pertinence) minimum.

- *Nombre de chemins passant par les c*
- *Les poids sur les noeuds, voir les arêtes*
- *Signature des chemins ?*

Voir du coté de « betweenness centrality », concept qui a l'air très proche.

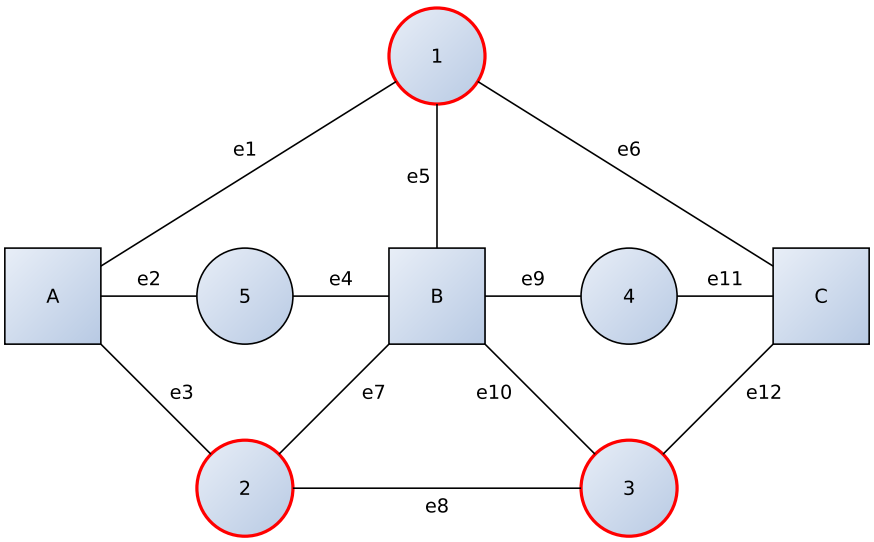

FIGURE 2.1 – Exemple C_I

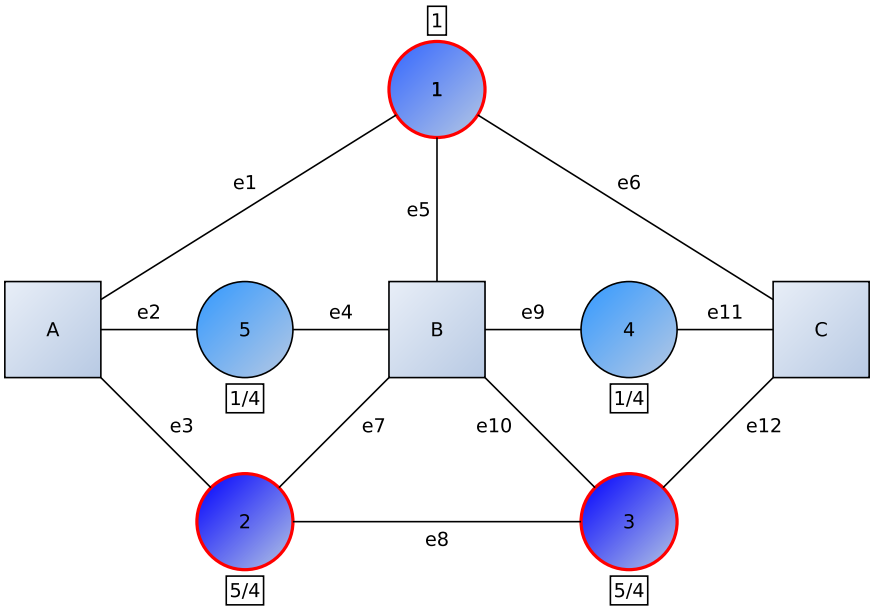

FIGURE 2.2 – Exemple (suite) C_I ordonné

Conclusion

Bibliographie

[1] Mathieu Lafourcade. *Lexique et analyse sémantique de textes - structures, acquisitions, calculs, et jeux de mots*. Hdr, Université Montpellier II - Sciences et Techniques du Languedoc, December 2011. URL `http://tel.archives-ouvertes.fr/tel-00649851`.

[2] Mathieu Lafourcade and Alain Joubert. Similitude entre les sens d'usage d'un terme dans un réseau lexical. *Traitement Automatique des Langues*, 50(1) :179–200, 2009. URL `http://hal-lirmm.ccsd.cnrs.fr/lirmm-00507777`.

[3] Alain Joubert, Mathieu Lafourcade, Didier Schwab, and Michael Zock. Évaluation et consolidation d'un réseau lexical grâce à un assistant ludique pour le " mot sur le bout de la langue ". In *TALN'11 : Traitement Automatique des Langues Naturelles*, pages 295–306, Montpellier, France, June 2011. URL `http://hal-lirmm.ccsd.cnrs.fr/lirmm-00832991`.

[4] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25 : 163–177, 2001.

[5] L.C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40 :35–41, 1977.