Comparaison de textes: quelques approches...

Elsa Negre

Laboratoire d'Analyses et Modélisation de Systèmes pour l'Aide à la Décision
UMR 7243

CAHIER DU LAMSADE

338
Avril 2013

Comparaison de textes
quelques approches…

Elsa Negre

Date d'impression : 2 avril 2013

Table des matières

Chapitre 1

Introduction

Dans ce document, nous présentons certaines approches permettant de comparer des textes. Nous parlerons par la suite de similarité entre textes ou entre documents. Les approches présentées ont été sélectionnées pour répondre au mieux au contexte (détaillé dans la section suivante). Ainsi, ce document ne prétend pas donner une liste exhaustive de toutes les méthodes existantes, mais tente de donner un aperçu des méthodes les plus utilisées dans le contexte de notre étude.

1.1 Contexte

L'étude présentée ici consiste à donner des pistes sur des mesures de similarité entre documents afin de prendre la meilleure décision de développement possible. Il s'agit de comparer des textes. Ces textes sont de petites tailles, à savoir, un maximum de 500 caractères. Ces textes ont été saisis grâce à un smartphone. Les calculs de similarité pourront être réalisés sur un serveur distant mais devront quelques fois être réalisés très rapidement. Les résultats devront donc être pertinents et être obtenus le plus rapidement possible. Nous souhaitons donc déterminer, pour un texte donné d_a, un ensemble de textes D_{min} similaires, i.e. qui sont les plus proches de d_a. Plus formellement, soit D un ensemble de n documents textuels, tel que pour chaque document $\forall i \in [1..n], d_i \in D$, le nombre de caractères contenus n'excède pas 500. Pour un document donné d_a, nous souhaitons obtenir l'ensemble des documents D_{min} qui minimisent la mesure de similarité avec d_a, i.e., $D_{min} = \{\forall d_a \in D, \forall d_b, d_c \in \{D \setminus d_a\}, sim(d_a, d_b) < sim(d_a, d_c)\}$.

1.2 Similarité entre textes

Évaluer la similarité entre documents textuels est une des problématiques importantes de plusieurs disciplines comme l'analyse de données textuelles, la recherche d'information ou l'extraction de connaissances à partir de données textuelles (Text Mining). Dans chacun de ces domaines, les similarités sont utilisées pour différents traitements :

- en analyse de données textuelles, les similarités sont utilisées pour la description et l'exploration de données ;
- en recherche d'information, l'évaluation des similarités entre documents et requêtes est utilisée pour identifier les documents pertinents par rapport à des besoins d'information exprimés par les utilisateurs ;

– en Text Mining, les similarités sont utilisées pour produire des représentations synthétiques de vastes collections de documents.

Les techniques mises en oeuvre pour calculer les similarités varient bien évidemment selon les disciplines, mais elles s'intègrent cependant le plus souvent dans une même approche générale en deux temps :

1. Les documents textuels sont d'abord associés à des représentations spécifiques qui vont servir de base au calcul des similarités. Bien que la nature précise des représentations utilisées dépende fortement du domaine d'application, il faut noter que, presque dans tous les cas, les documents sont représentés sous la forme d'éléments d'un espace vectoriel de grande dimension.
2. Un modèle mathématique est choisi pour mesurer les similarités.

1.2.1 Similarité syntaxique

En mathématiques et en informatique, une mesure permettant de comparer des documents textuels, consiste à comparer des chaines de caractères. C'est une métrique qui mesure la similarité ou la dissimilarité entre deux chaînes de caractères. Par exemple, les chaînes de caractères "Sam" et "Samuel" peuvent être considérées comme similaires. Une telle mesure sur les chaines de caractères fournit une valeur obtenue algorithmiquement.

Parmi de telles mesures de similarité, citons par exemple, la distance de Levenshtein (ou distance d'édition), le coefficient de Dice, l'indice de Jaccard, la distance euclidienne, le cosinus, ...

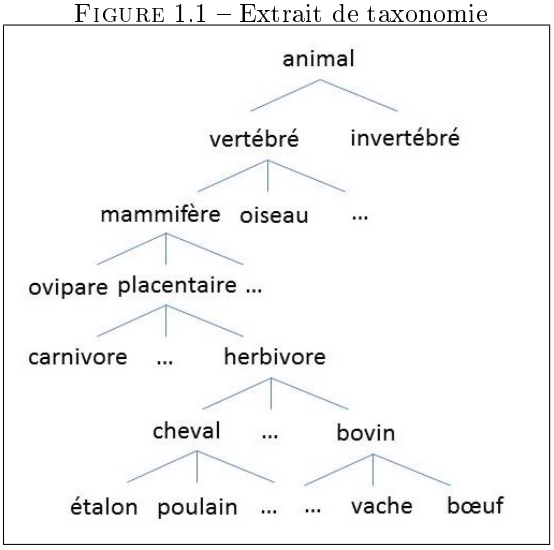
1.2.2 Similarité sémantique

La similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification / contenu sémantique.

Concrètement, cela peut être réalisé en définissant une similitude topologique, par exemple, en utilisant des ontologies pour définir une distance entre les mots, ou en définissant une similitude statistique, par exemple en utilisant un modèle d'espace vectoriel pour corréler les termes et les contextes à partir d'un corpus de texte approprié (co-occurrence).

Parmi de telles mesures de similarité, citons par exemple, Resnik, LSA (Analyse sémantique latente), ESA (Analyse sémantique explicite), ...

FIGURE 1.1 – Extrait de taxonomie



1.3 Exemple

A première vue, les trois extraits de textes présentés dans cette section n'ont pas vraiment de points communs. Dans la suite de ce rapport, nous allons pouvoir déterminer la proximité, i.e., la similarité, entre ces différents extraits, selon différentes techniques de calcul de similarité entre documents. Chaque proposition sera illustrée [1] par un exemple. Par conséquent, dans la suite de ce rapport, les documents utilisés seront les suivants :

d_1 : extrait de "Les yeux d'Elsa" d'Aragon

Tes yeux sont si profonds qu'en me penchant pour boire
J'ai vu tous les soleils y venir se mirer
S'y jeter à mourir tous les désespérés
Tes yeux sont si profonds que j'y perds la mémoire

d_2 : extrait de "Le Fou et la Vénus" de Baudelaire

Quelle admirable journée ! Le vaste parc se pâme sous l'œil brûlant du soleil, comme la jeunesse sous la domination de l'Amour.

d_3 : proverbe arabe

A quoi sert la lumière du soleil, si on a les yeux fermés.

De plus, nous allons considérer l'extrait de taxonomie illustré Figure 1.1.

1. Chaque exemple sera donné à titre indicatif et aura été réalisé manuellement.

Chapitre 2

Similarité syntaxique

2.1 Présentation

Comme indiqué dans la section 1.2.1, une mesure de similarité syntaxique permet de comparer des documents textuels en se basant sur les chaines de caractères qui les composent. Par exemple, les chaînes de caractères "voiture" et "voiturier" peuvent être considérées comme très proches, alors que "voiture" et "automobile" pourront être considérées comme très différentes.

Dans ce chapitre, nous présentons les mesures de similarité syntaxique les plus utilisées, en passant par la représentation vectorielle d'un document.

2.2 Modèle d'espace vectoriel

Afin de réduire la complexité des documents et de faciliter leur manipulation, il faut transformer chaque document, i.e. sa version textuelle intégrale, en un vecteur qui décrit le contenu du document. La représentation d'un ensemble de documents sous forme de vecteurs dans un espace vectoriel commun est connu sous le nom de modèle d'espace vectoriel (*vector space model*). En recherche d'information, dans un modèle d'espace vectoriel, les documents sont représentés comme des vecteurs de caractéristiques représentant les termes qui apparaissent dans la collection. On parle aussi de 'sacs de mots' où les mots sont considérés comme indépendants et où l'ordre est sans importance. La valeur de chaque caractéristique est appelé le poids du terme et est en général une fonction de fréquence de termes dans le document. Par conséquent, en utilisant la fréquence de chaque terme comme un poids, les termes qui apparaissent le plus fréquemment sont plus importants et donc descriptifs du document. La représentation d'un document sous forme vectorielle se déroule en 2 étapes :

1. extraire les termes pertinents du document ;

2. calculer les poids des termes restants.

2.2.1 Extraction des termes pertinents

Il s'agit de pré-traiter le texte des documents textuels en supprimant les mots-vides, la ponctuation et les éventuels 'retours-chariots', de lemmatiser [1] le texte et de le segmenter.

Exemple : A partir des documents d_1 et d_2 donnés section 1.3, après élimination de la ponctuation, élimination des mots-vides [2] et lemmatisation [3], l'ensemble de termes pertinents pour d_1 est : {yeux, si, profond, pench, boir, v, tou, soleil, ven, mir, jet, mour, desesp, perd, memo} ; pour d_2 : { admir, jour, vaste, parc, pam, sous, oeil, brul, soleil, comme, jeune, domin, amour}.

2.2.2 Calcul des poids

Le poids de chaque terme dans un document peut être obtenu de différentes manières : booléenne, fréquence des termes, tf-idf (*Term frequency - Inverse Document Frequency*).

Méthode booléenne
De manière booléenne, si un terme existe dans un document alors la valeur qui lui correspond vaut 1, sinon 0. L'approche booléenne est utilisée lorsque chaque terme est d'égale importance et s'emploie uniquement lorsque les documents sont de petites tailles.

Exemple : L'ensemble des termes pertinents extraits de tous les documents de la collection est : {yeux, si, profond, pench, boir, v, tou, soleil, ven, mir, jet, mour, desesp, perd, memo, admir, jour, vaste, parc, pam, sous, oeil, brul, comme, jeune, domin, amour}. Les représentations vectorielles de manière booléenne de d_1 et d_2 sont :

	yeux	si	profond	pench	boir	v	tou	soleil	ven	mir	jet	mour	desesp	perd	memo	admir
d_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
d_2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

jour	vaste	parc	pam	sous	oeil	brul	comme	jeune	domin	amour
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1

Fréquence des termes
Pour la fréquence des termes, le poids d'un terme est obtenu en comptant les occurrences du terme dans le document : $tf_{i,j}$ représente donc la fréquence du terme i dans le document j.

Exemple : Les représentations vectorielles de d_1 et d_2 avec la fréquence des termes sont :

	yeux	si	profond	pench	boir	v	tou	soleil	ven	mir	jet	mour	desesp	perd	memo	admir
d_1	$\frac{2}{27}$	$\frac{2}{27}$	$\frac{2}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{2}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	0
d_2	0	0	0	0	0	0	0	$\frac{1}{27}$	0	0	0	0	0	0	0	$\frac{1}{27}$

jour	vaste	parc	pam	sous	oeil	brul	comme	jeune	domin	amour
0	0	0	0	0	0	0	0	0	0	0
$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{2}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$	$\frac{1}{27}$

1. La lemmatisation du contenu d'un texte permet de regrouper les mots d'une même famille. Chacun des mots d'un contenu se trouve ainsi réduit en une entité appelée lemme (forme canonique). La lemmatisation regroupe les différentes formes que peut revêtir un mot, soit : le nom, le pluriel, le verbe à l'infinitif, ...

2. Les mots considérés vides, ici, sont : *pour, le, la, les, l',..., un, une, des, ..., y, à, se, s', les dérivés du verbe être, les dérivés du verbe avoir, mon, ton, son, mes, tes, ..., qu', que, qui, ...*

3. Quelques exemples de lemmatisation pour cet exemple : *yeux* → *yeux*, *profonds* → *profond*, *jeter* → *jet*, *desesperes* → *desesp*, *memoire* → *memo*, *domination* → *domin*, ... L'algorithme de Porter [Porter, 1980] est le plus utilisé.

Tf-Idf
Le tf-idf permet, quant à lui, d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection. Le poids augmente proportionnellement avec le nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans la collection. Ainsi, la fréquence inverse du document (idf) est une mesure de l'importance du terme dans l'ensemble des documents. Dans le cas du tf-idf, elle vise à donner un poids plus important aux termes les moins fréquents, considérés comme les plus discriminants. Il s'agit de calculer le logarithme de l'inverse de la proportion de documents qui contiennent le terme : $idf_i = log\frac{|D|}{|\{d_j : t_i \in d_j\}|}$ où $|D|$ est le nombre total de documents et $|\{d_j : t_i \in d_j\}|$ est le nombre de documents où le terme t_i apparait. Finalement, le poids s'obtient en multipliant les deux mesures : $tfidf_{i,j} = tf_{i,j}.idf_i$.

Exemple : Notre exemple contient deux documents, donc $|D| = 2$. Pour le terme "yeux", seul le document d_1 le contient. Par conséquent, $idf_{yeux} = log(\frac{2}{1}) \approx 0,301$. De même, le terme "soleil" apparait dans les deux documents, donc $idf_{soleil} = log(\frac{2}{2}) = 0$. Le calcul du tf-idf du terme "yeux" est donc : $tfidf_{yeux,d_1} = tf_{yeux,d_1}.idf_{yeux} = \frac{2}{27}.log\frac{2}{1} \approx 0,0222$ pour le document d_1 et $tfidf_{yeux,d_2} = tf_{yeux,d_2}.idf_{yeux} = 0.log\frac{2}{1} = 0$ pour le document d_2. Celui du terme "soleil" est donc : $tfidf_{soleil,d_1} = tf_{soleil,d_1}.idf_{soleil} = \frac{1}{27}.log\frac{2}{2} = 0$ pour le document d_1 et $tfidf_{soleil,d_2} = tf_{soleil,d_2}.idf_{soleil} = \frac{1}{27}.log\frac{2}{2} = 0$ pour le document d_2.

Les représentations vectorielles de d_1 et d_2 avec le tf-idf sont :

	yeux	si	profond	pench	boir	v	tou	soleil	ven	mir	jet	mour	desesp	perd	memo	admir
d_1	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0
d_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,01

	jour	vaste	parc	pam	sous	oeil	brul	comme	jeune	domin	amour
	0	0	0	0	0	0	0	0	0	0	0
	0,01	0,01	0,01	0,01	0,02	0,01	0,01	0,01	0,01	0,01	0,01

2.3 Mesures de similarité existantes

Une mesure de similarité est, en général, une fonction qui quantifie le rapport entre deux objets, comparés en fonction de leurs points de ressemblance et de dissemblance. Les deux objets comparés sont, bien entendu, de même type.

2.3.1 Métriques

Toutes les mesures de similarité ne sont pas des métriques. Pour être une métrique, une mesure d doit satisfaire les 4 conditions suivantes :
Soit x, y et z, trois éléments d'un ensemble, et soit $d(x, y)$ la distance entre x et y.
- Positivité : $d(x, y) \geq 0$.
- Principe d'identité des indiscernables : $d(x, y) = 0 \equiv x = y$.
- Symétrie : $d(x, y) = d(y, x)$.
- Inégalité triangulaire : $d(x, z) \leq d(x, y) + d(y, z)$

2.3.2 Similarité Cosinus

La similarité cosinus est fréquemment utilisée [Baeza-Yates and Ribeiro-Neto, 1999] en tant que mesure de ressemblance entre deux documents d_1 et d_2. Il s'agit de calculer le cosinus de l'angle entre les représentations vectorielles des documents à comparer. La similarité obtenue $sim_{cosinus}(d_1, d_2) \in [0, 1]$.

$$sim_{cosinus}(d_1, d_2) = \frac{\vec{d_1}.\vec{d_2}}{\|\vec{d_1}\|\|\vec{d_2}\|}$$

Exemple : Par souci de lisibilité et de facilité des calculs, nous nous limitons ici à des vecteurs de taille 6 contenant les termes : {yeux, profond, soleil, memo, sous, oeil}. Les représentations vectorielles de d_1, d_2 et d_3 avec le tf sont :

	yeux	profond	soleil	memo	sous	oeil
d_1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{6}$	0	0
d_2	0	0	$\frac{1}{6}$	0	$\frac{1}{3}$	$\frac{1}{6}$
d_3	$\frac{1}{6}$	0	$\frac{1}{6}$	0	0	0

Nous avons donc $sim_{cosinus}(d_1, d_2) = \frac{\vec{d_1}.\vec{d_2}}{\|\vec{d_1}\|\|\vec{d_2}\|} = \frac{\frac{1}{36}}{\sqrt{\frac{10}{36}}\cdot\sqrt{\frac{1}{6}}} \approx 0,129$. De la même manière, $sim_{cosinus}(d_1, d_3) \approx 0,447$ et $sim_{cosinus}(d_2, d_3) \approx 0,288$. Par conséquent, selon la similarité cosinus, d_1 et d_3 sont les plus similaires.

2.3.3 Coefficient de corrélation de Pearson

Le coefficient de corrélation de Pearson calcule la similarité entre deux documents d_1 et d_2 comme le cosinus de l'angle entre leurs représentations vectorielles centrées-réduites. La similarité obtenue $sim_{pearson}(d_1, d_2) \in [-1, 1]$.

$$sim_{pearson}(d_1, d_2) = sim_{cosinus}(d_1 - \overline{d_1}, d_2 - \overline{d_2})$$

où $\overline{d_1}$ (resp. $\overline{d_2}$) représente la moyenne de d_1 (resp. d_2).

Exemple : En gardant des vecteurs de taille 6, la moyenne pour d_1, $\overline{d_1} = \frac{1}{6}$, pour d_2, $\overline{d_2} = \frac{1}{9}$ et pour d_3, $\overline{d_3} = \frac{1}{18}$, nous avons $sim_{pearson}(d_1, d_2) = sim_{cosinus}(d_1 - \overline{d_1}, d_2 - \overline{d_2}) = \frac{-\frac{1}{12}}{\sqrt{\frac{4}{36}}\cdot\sqrt{\frac{30}{324}}} \approx -0,821$. De la même manière, $sim_{pearson}(d_1, d_3) \approx 0,433$ et $sim_{pearson}(d_2, d_3) \approx -0,158$. Par conséquent, selon le coefficient de Pearson, d_2 et d_3 sont les plus similaires.

2.3.4 Distance euclidienne

La distance euclidienne calcule la similarité entre deux documents d_1 et d_2 comme la distance entre leurs représentations vectorielles ramenées à un seul point.

$$sim_{euclidienne}(d_1, d_2) = \| \vec{d_1} - \vec{d_2} \| = \sqrt{\sum_{i=1}^{n}(d_{1_i} - d_{2_i})^2}$$

où n est le nombre total de termes représentés, i.e. la taille des vecteurs.

Exemple : En gardant des vecteurs de taille 6, nous avons $sim_{euclidienne}(d_1, d_2) = \| \vec{d_1} - \vec{d_2} \| = \sqrt{\sum_{i=1}^{n}(d_{1_i} - d_{2_i})^2} = \sqrt{\frac{10}{27}} \approx 0,608$. De la même manière, $sim_{euclidienne}(d_1, d_3) \approx 0,408$ et $sim_{euclidienne}(d_2, d_3) \approx 0,408$. Par conséquent, selon la distance euclidienne, d_3 est à la même distance de d_1 que de d_2.

2.3.5 Coefficient de Jaccard

L'indice de Jaccard ou coefficient de Jaccard [Jaccard, 1901] est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles. Les documents d_1 et d_2 sont donc représentés, non pas comme des vecteurs, mais comme des ensembles de termes. La similarité obtenue $sim_{jaccard}(d_1, d_2) \in [0, 1]$.

$$sim_{jaccard}(d_1, d_2) = \frac{\|d_1 \cap d_2\|}{\|d_1 \cup d_2\|}$$

Il est aussi possible d'utiliser la représentation vectorielle.

$$sim_{jaccard}(d_1, d_2) = \frac{\vec{d_1}.\vec{d_2}}{\|\vec{d_1}\|\|\vec{d_2}\| - \vec{d_1}.\vec{d_2}}$$

Exemple : En gardant des vecteurs de taille 6, nous avons $sim_{jaccard}(d_1, d_2) = \frac{\vec{d_1}.\vec{d_2}}{\|\vec{d_1}\|\|\vec{d_2}\| - \vec{d_1}.\vec{d_2}} = \frac{\frac{1}{36}}{\sqrt{\frac{10}{36}} \cdot \sqrt{\frac{1}{6}} - \frac{1}{36}} \approx 0,148$. De la même manière, $sim_{jaccard}(d_1, d_3) \approx 0,809$ et $sim_{jaccard}(d_2, d_3) \approx 0,405$. Par conséquent, selon le coefficient de Jaccard, d_1 et d_3 sont les plus similaires.

2.3.6 Distance (d'édition) de Levenshtein

La distance de Levenshtein [4] [Levenshtein, 1966] calcule la similarité entre les représentations sous forme de chaines de caractères des documents d_1 et d_2. Il s'agit du coût minimal, i.e. du nombre minimal d'opérations d'édition, pour transformer d_1 en d_2. Les opérations sont les suivantes :
– substitution d'un caractère de d_1 en un caractère de d_2,
– ajout dans d_1 d'un caractère de d_2,
– suppression d'un caractère de d_1.
Pour obtenir la distance de Levenshtein $sim_{levenshtein}(d_1, d_2)$ entre les documents d_1 et d_2, il s'agit d'associer à chacune de ces opérations un coût. Le coût des opérations est toujours égal à 1, sauf dans le cas d'une substitution de caractères identiques. Notons que cette distance a été étendue pour prendre en compte la grammaire, la phonétique, ...

Exemple : Par souci de lisibilité, nous nous limitons ici aux trois premiers mots de chaque document. Ainsi, nous avons, pour d_1 : "Tes yeux sont", pour d_2 : "Quelle admirable journée" et pour d_3 : "A quoi sert". Le tableau suivant indique les opérations nécessaires pour transformer d_1 en d_2 et ainsi obtenir la distance de Levenshtein entre d_1 et d_2.

d_1	t	e	s		y	e	u	x		s	o	n
d_2	q	u	e	l	l	e		a	d	m	i	r
opération	$t \rightarrow q$	$e \rightarrow u$	$s \rightarrow e$	ajout l	$y \rightarrow l$	/	suppr u	$x \rightarrow a$	ajout d	$s \rightarrow m$	$o \rightarrow i$	$n \rightarrow r$

d_1	t												
d_2	a	b	l	e		j	o	u	r	n	é	e	
opération	$t \rightarrow a$	ajout b	ajout l	ajout e	ajout espace	ajout j	ajout o	ajout u	ajout r	ajout n	ajout é	ajout e	

Ce qui fait 23 opérations de coût égal à 1 pour transformer d_1 en d_2. Par conséquent, $sim_{levenshtein}(d_1, d_2) = 23$. De la même manière, $sim_{levenshtein}(d_1, d_3) = 9$ et $sim_{levenshtein}(d_2, d_3) = 23$. Par conséquent, selon la distance de Levenshtein, d_1 et d_3 sont les plus similaires.

4. La distance de Levenshtein est une mesure permettant l'appariement approximatif de chaines de caractères (*approximate string matching*) [Navarro, 2001]

2.3.7 Indice de Dice

L'indice de Dice mesure la similarité entre deux documents d_1 et d_2 en se basant sur le nombre de termes communs à d_1 et d_2.

$$sim_{dice}(d_1, d_2) = \frac{2N_c}{N_1+N_2}$$

où N_c est le nombre de termes communs à d_1 et d_2, et N_1 (resp. N_2) est le nombre de termes de d_1 (resp. d_2).

Exemple : Les documents d_1 et d_2 ont deux termes en commun, à savoir "se" et "la". Le document d_1 contient 36 mots. Le document d_2 contient 21 mots. Ainsi, l'indice de Dice entre les documents d_1 et d_2 vaut : $sim_{dice}(d_1, d_2) = \frac{2N_c}{N_1+N_2} = \frac{2*2}{36+21} \approx 0,07$. De la même manière, $sim_{dice}(d_1, d_3) \approx 0,204$ et $sim_{dice}(d_2, d_3) \approx 0,117$. Par conséquent, selon l'indice de Dice, d_1 et d_3 sont les plus similaires.

2.4 Outils

Toutes ces mesures ont déjà été développées sur de nombreux supports. Cette section fera l'objet de recherches plus approfondies lorsqu'une décision aura été prise quant à la solution à adopter.

2.5 Comparatif

2.5.1 Performances

Nous tentons ici d'identifier les performances de chaque mesure.

[Huang, 2008] et [Strehl et al., 2000] ont tous les deux montré que les performances de la similarité cosinus, du coefficient de Jaccard et du coefficient de Pearson sont très proches et qu'elles sont significativement meilleures que celles de la distance euclidienne. Cependant, [Bavi et al., 2010] fait apparaitre que plus le document est de petite taille, meilleurs sont les résultats obtenus avec la distance euclidienne, tandis qu'ils sont plus mauvais avec la similarité cosinus ou avec le coefficient de Jaccard.

Sachant que l'indice de Dice est fonction du Jaccard [5], nous pouvons penser qu'ils ont des performances similaires. La distance de Levenshtein est largement utilisée en linguistique et en bioinformatique ainsi que pour la reconnaissance de blocs de textes contenants des erreurs isolées. Malheureusement, le temps de calcul (complexité), lorsqu'on l'applique à deux séquences d'approximativement la même taille, n, est $O(n^2)$. Cela est un obstacle dans de nombreuses applications pratiques [Baake et al., 2006].

2.5.2 Avantages et Inconvénients

Nous tentons ici de lister (de manière non exhaustive) les avantages et les inconvénients, non pas de chaque mesure, mais du type d'approche à laquelle les mesures appartiennent.

5. Soit D l'indice de Dice et J le coefficient de Jaccard, nous avons $D = \frac{2J}{1+J}$.

Avantages et inconvénients liés au modèle vectoriel

Avantages :
- Quelque soit la technique utilisée, basée sur le modèle vectoriel, a de fait, le même format initial, à savoir, la représentation vectorielle ;
- Les techniques basées sur le modèle vectoriel sont faciles à développer, il s'agit uniquement de calcul vectoriel.

Inconvénients :
- Des mots identiques considérés comme peu pertinents peuvent parfois trop influer sur la valeur de la similarité. Par exemple, pour les phrases "Tout *est bien* qui finit *bien*" et "C'*est* notre seul *bien*", le terme "est" n'est pas vraiment pertinent et pourtant, il va avoir un poids certain.

Notons cependant que la lemmatisation, l'élimination des mots-vides et le tf-idf permettent de pallier cet inconvénient.

Avantages et inconvénients liés aux approches syntaxiques

Avantages :
- Les techniques basées sur l'approche syntaxique ne laissent pas de place aux exceptions ;
- Elles sont donc facilement automatisables.

Inconvénients :
- Par définition, les techniques basées sur l'approche syntaxique ne prennent pas en compte la sémantique. Par exemple, il est difficile de trouver une forte similarité entre "Je possède un chien" et "J'ai un animal". Dans le contexte de notre étude, plusieurs mots peuvent être utilisés pour parler du même objet. Par conséquent, la prise en compte de la sémantique semble importante.
- Les relations syntaxiques sont ignorées. Par exemple, aucune différence n'est faite entre "Pierre aime Marie" et "Marie aime Pierre". Dans le contexte de notre étude, les relations syntaxiques peuvent influer sur la pertinence. Par conséquent, il faudrait trouver un moyen d'incorporer des techniques d'analyse de variation du texte.
- De même, les rôles sémantiques sont ignorés. Par exemple, dans "La société A achète la société B" et "La société B a été achetée par la société A", seule la forme verbale change. Cela peut engendrer des problèmes de pertinence. Une proposition serait peut-être d'analyser les classes verbales.
- Les problèmes liés aux négations (par exemple, "Je suis malade" et "Je ne suis pas malade") et aux antinomies semblent encore difficiles à pallier.

Chapitre 3

Similarité sémantique

3.1 Présentation

Comme indiqué dans la section 1.2.2, une mesure de similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification / contenu sémantique.

Il est à noter que la distance sémantique peut être de deux sortes : la similarité sémantique et la parenté sémantique. La première est un sous-ensemble de la seconde, mais les deux termes peuvent être utilisés indifféremment dans certains contextes, ce qui rend encore plus important d'être conscient de leur distinction. Deux concepts sont considérés comme sémantiquement similaires s'il y a une synonymie, hyponymie[1], antonymie, ou troponymie[2] entre eux (Exemples : MEDECIN-CHIRURGIEN, SOMBRE-CLAIR). Deux sens de mots sont considérés comme sémantiquement liés s'il existe au moins une relation lexico-sémantique entre eux - classique ou non classique (Exemples : CHIRURGIEN-SCALPEL, ARBRE-OMBRE) [Mohammad and Hirst, 2012].

Les mesures de similarité de textes ont été utilisées dans de nombreux domaines. Par exemple, pour la classification de textes ([Rocchio, 1971]), la désambiguïsation du sens des mots ([Lesk, 1986]), la traduction automatique ([Papineni et al., 2002])...

À quelques exceptions près, l'approche classique pour trouver la similarité entre deux segments de texte, est d'utiliser une méthode simple de concordance lexicale, et de calculer un score de similarité basé sur le nombre d'unités lexicales qui se produisent dans les deux segments. Des améliorations ont été apportées à cette méthode simple qui consistent à retirer les mots-vides, à ne considérer que la plus longue sous-suite, ou encore à pondérer ou normaliser ([Salton, 1997]). Ces méthodes de similarité lexicale ne peuvent pas toujours identifier la similarité sémantique des textes. Par exemple, il y a une similitude évidente entre les segments de texte "*Je possède un chien*" et "*j'ai un animal*", mais la plupart des mesures de similarité textuelles vont échouer à identifier tout type de connexion entre ces textes.

1. Relation sémantique hiérarchique d'un lexème à un autre selon laquelle l'extension du premier terme est incluse dans l'extension du second. Le premier terme est dit hyponyme de l'autre. *Haut-de-forme* est un hyponyme de *chapeau* et *chapeau* est un hyponyme de *coiffure*.

2. Relation sémantique entre deux verbes, l'un décrivant de manière plus précise l'action de l'autre. Le premier verbe est dit troponyme du second.

Il existe des mesures de similarité sémantique qui tentent de réussir en utilisant des approches qui sont, soit fondées sur la connaissance ([Wu and Palmer, 1994, Leacock and Chodorow, 1998], ...) ou sur un corpus ([Deerwester et al., 1990], ...). Ces mesures ont été appliquées avec succès à des tâches de traitement du langage comme, par exemple, la désambiguïsation du sens des mots ([Patwardhan et al., 2003]).

3.2 Mesures existantes

Dans cette section, nous dissocions les approches vectorielles, les approches topologiques et les approches statistiques.

3.2.1 Approches vectorielles

Vecteurs sémantiques

L'idée consiste à déterminer la sémantique d'un mot en consultant les autres termes utilisés à ses côtés dans des phrases. Une manière simple de le faire est d'utiliser des vecteurs pour représenter le sens des mots, et d'utiliser ensuite des mesures de similarité vectorielles (comme pour la similarité syntaxique). Le plus difficile est d'obtenir de tels vecteurs. Il faut donc construire un ensemble de vecteurs pour chaque mot dans le dictionnaire utilisé. Les vecteurs sont définis dans un espace vectoriel orthogonal à n dimensions où chaque base se voit attribuer un mot de vocabulaire unique (donc chaque entrée du dictionnaire a une base dans l'espace vectoriel). Pour chaque mot du dictionnaire, on détermine un vecteur dans cet espace, où la composante du vecteur pour chaque base est le nombre d'occurrences du mot dans la base qui le représente où il apparaît dans le contexte du mot pour lequel un vecteur a été construit. Le mot "Contexte" ici peut être vu au sens large.

Exemple : Supposons que nous disposions d'un dictionnaire contenant uniquement 3 mots : *cheval*, *poulain* et *vache*. De plus, poulain apparait avec cheval dans 12 documents, et avec vache dans seulement 3. Nous considérons un espace vectoriel avec uniquement vache et cheval en tant que vecteurs de base, dans lesquels poulain apparaitrait 3 fois dans la base "vache" et 12 fois dans la base "cheval". Si nous utilisons la similarité cosinus, nous avons donc : $sim_{semvec}(poulain, cheval) = \frac{3*0+12*1}{\sqrt{3^2+12^2}*1} = \frac{12}{\sqrt{153}} \approx 0,97$ et $sim_{semvec}(poulain, vache) = \frac{3*1+12*0}{\sqrt{3^2+12^2}*1} = \frac{3}{\sqrt{153}} \approx 0,24$ Ainsi, poulain est plus proche de cheval que de vache puisqu'il apparait plus souvent dans le même contexte que cheval.

Bi-clustering

La classification double ou co-clustering ou bi-clustering est une technique d'exploration de données non-supervisée permettant de segmenter simultanément les lignes et les colonnes d'une matrice. Étant donné un ensemble de r lignes à c colonnes (c'est-à-dire une matrice $r \times c$), l'algorithme de bi-clustering génère des bi-clusters - un sous-ensemble de lignes qui présentent un comportement similaire sur un sous-ensemble de colonnes, ou vice versa.

Le bi-clustering est utilisé dans le domaine de la fouille de texte, où il est populairement connu en tant que co-clustering [Bisson and Hussain, 2008]. Les corpus de textes sont représentés sous une forme vectorielle : comme une matrice D dont les lignes sont les documents et les colonnes sont les mots du dictionnaire. Les éléments D_{ij} de la matrice désignent l'occurrence du mot j dans le document i. Les algorithmes de

bi-clustering sont ensuite appliqués pour découvrir des blocs dans D qui correspondent à un groupe de documents (lignes) caractérisé par un groupe de mots (colonnes).

[Bisson and Hussain, 2008] ont proposé une approche qui utilise la similarité entre les mots et la similarité entre les documents pour segmenter la matrice. Leur méthode (connue sous le nom $\chi - Sim$, pour similarité croisée) est basée sur la recherche de similarité document-document et de similarité mot-mot, puis utilise les méthodes classiques de classification. Au lieu de regrouper explicitement les lignes et les colonnes alternativement, les auteurs considèrent des occurrences de mots d'ordre supérieur, i.e., en tenant compte des documents dans lesquels ils apparaissent. Ainsi, la similarité entre deux mots est calculée sur la base des documents dans lesquels ils apparaissent ainsi que des documents dans lesquels des mots similaires apparaissent. L'idée est que deux documents sur le même sujet ne contiennent pas nécessairement le même jeu de mots, mais un sous-ensemble des mots et d'autres mots similaires qui sont caractéristiques de ce sujet. Cette approche d'utilisation de similarités d'ordre supérieur prend en considération la structure sémantique latente de l'ensemble du corpus et génère ainsi une meilleure classification des documents et des mots.

Plus formellement, à partir de la matrice documents/termes D (où le vecteur ligne $d_{i:}$ de taille c décrit le document i et le vecteur colonne $d_{:j}$ de taille r décrit le mot j), il s'agit de déterminer les matrices SR (matrice de similarité pour les documents) et SC (matrice de similarité pour les termes). Classiquement, la similarité entre deux documents est une fonction sur les termes communs. Ainsi, nous avons l'équation : $sim(d_{i:}, d_{:j}) = F_s(d_{i1}, d_{j1}) + ... + F_s(d_{ic}, d_{jc})$ où F_s est une fonction de similarité. Initialement, on suppose que la matrice SC est initialisée à 1, i.e., $sc_{i,i} = 1$. Notre équation devient donc $sim(d_i, d_j) = F_s(d_{i1}, d_{j1}).sc_{11} + ... + F_s(d_{ic}, d_{jc}).sc_{cc}$. Cette équation est ensuite généralisée pour prendre en compte toutes les paires de mots possibles. Réciproquement, la similarité entre deux mots est fonction des documents communs dans lesquels ils apparaissent. Ainsi, les équations pour calculer les similarités sont dépendantes : pour obtenir SR et SC, il faut utiliser itérativement et alternativement chaque équation et mettre à jour les valeurs pour s'en servir dans les itérations suivantes.

Exemple : Dans le tableau ci-dessous, les documents d_1 et d_2 n'ont aucun mot w_i en commun. Avec une mesure de similarité syntaxique, leur similarité est égale à zéro (ou la distance qui les sépare est maximale). Pourtant, on peut observer que d_1 et d_2 partagent des mots avec d_3, ce qui signifie que les mots w_2 et w_3 ont quelques similitudes dans l'espace des documents. Avec une approche de bi-clustering, il est possible d'associer une similarité entre d_1 et d_2, qui sera bien sûr moins flagrante que celles entre d_1 et d_3 ou entre d_2 et d_3 mais elle sera néanmoins non nulle.

M	w_1	w_2	w_3	w_4
d_1	1	1	0	0
d_2	0	0	1	1
d_3	0	1	1	0

3.2.2 Approches topologiques (ou knowledge-based)

Les approches de similarité de mots basées sur la connaissance s'appuient sur un réseau sémantique de mots, tel que WordNet [Fellbaum, 1998]. Etant donnés deux mots, leur similarité peut être estimée à partir de leurs positions relatives dans la hiérarchie de la base de connaissances. En effet, la structure de la base est un arbre où chaque noeud est un concept (par exemple, un chat), ses enfants sont les hyponymes du concept (i.e., 'X' est un hyponyme de 'Y' si 'X est un Y' est vrai), et ses parents sont ses hyperonymes (i.e., 'X' est un hyperonyme de 'Y' si 'Y est un X' est vrai). Les concepts peuvent être des noms, des verbes ou des adjectifs. Les mots ont des "*synsets*", qui sont des ensembles de concepts pour lesquels le mot peut correspondre (i.e., les concepts desquels le mot peut être synonyme). Enfin, il faut noter que les concepts sont de plus en plus abstraits et généraux lorsqu'on va vers la racine et qu'ils sont plus spécifiques lorsqu'on va vers les feuilles.

Wordnet est une base de connaissances ou taxonomie dont les concepts sont en anglais. Cependant, une base similaire a été créée pour la langue française : WOLF (WordNet Libre du Français)[Sagot and Fišer, 2008].

Edge-based

L'approche basée sur les arcs est une manière naturelle et directe d'évaluer la similarité sémantique dans une taxonomie. Il s'agit d'estimer la distance (e.g., longueur des arcs) entre les noeuds correspondants aux concepts / classes à comparer. Compte tenu de l'espace multidimensionnel des concepts, la distance conceptuelle peut facilement être mesurée par la distance géométrique entre les noeuds représentants les concepts. Évidemment, plus le chemin d'un noeud à l'autre est court, plus ils sont similaires.

Leacock et Chodorow Leacock et Chodorow [Leacock and Chodorow, 1998] ont utilisé une seule relation (hyponymie) et ont modifié la formule de longueur du chemin pour refléter le fait que les arcs les plus bas dans la hiérarchie d'hyponymie correspondent à la plus petite distance sémantique. Par exemple, les *synsets* relatifs à 'voiture de sport' et 'voiture' (bas dans la hiérarchie) sont beaucoup plus semblables que ceux relatifs au 'transport' et à 'l'instrumentation' (plus hauts dans la hiérarchie), bien que les deux paires de noeuds sont séparés par exactement un arc dans la hiérarchie. La similarité entre les deux concepts c_1 et c_2 est :

$$sim_{lch}(c_1, c_2) = -log\frac{longueur}{2D}$$

où *longueur* est la longueur du plus court chemin entre c_1 et c_2 (en terme de nombres de noeuds) et D est la profondeur/hauteur maximale de la taxonomie.

Exemple : A partir de la taxonomie de la figure 1.1, si nous considérons les concepts *vache*, *poulain* et *étalon*, nous avons : $sim_{lch}(vache, etalon) = -log\frac{4}{2*6} = 0,477$ et $sim_{lch}(poulain, etalon) = -log\frac{2}{2*6} = 0,778$. Par conséquent, *poulain* et *étalon* sont plus proches entre eux que de *vache*.

Wu et Palmer La métrique de similarité de Wu et Palmer [Wu and Palmer, 1994] mesure la profondeur de deux concepts donnés dans la taxonomie WordNet, et la profondeur de leur plus bas ancêtre commun (lowest common subsumer(LCS)) et les combine pour obtenir un score de similarité :

$$sim_{wup}(c_1, c_2) = \frac{2*profondeur(LCS)}{profondeur(c_1)+profondeur(c_2)}$$

Exemple : A partir de la taxonomie de la figure 1.1, si nous considérons les concepts *vache*, *poulain* et *étalon*, sachant que le LCS de *vache* et *étalon* est *herbivore* et le LCS de *étalon* et *poulain* est *cheval*, nous avons : $sim_{wup}(vache, etalon) = \frac{2*profondeur(herbivore)}{profondeur(vache)+profondeur(etalon)} = \frac{2*4}{6+6} = 0,666$ et $sim_{wup}(poulain, etalon) = 0,833$. Par conséquent, *poulain* et *étalon* sont plus proches entre eux que de *vache*.

Node-based (ou information content-based)

Une approche basée sur les noeuds pour déterminer la similarité conceptuelle est appelée une approche *information content-based* [Resnik, 1995]. Étant donné un espace multidimensionnel où un noeud représente

un concept unique composé d'un certain nombre d'informations, et où un arc représente une association directe entre deux concepts, la similarité entre deux concepts est la mesure dans laquelle ils partagent des informations en commun. Compte tenu de cette notion de structure hiérarchique / espace de classes, ces informations communes peuvent être identifiées comme un noeud/concept spécifique qui englobe les deux dans la hiérarchie. Plus précisément, cette super-classe devrait être la première classe en haut de la hiérarchie qui englobe les deux classes. La valeur de similarité est définie comme la valeur du contenu de l'information de cette super-classe. La valeur du contenu de l'information d'une classe est ensuite obtenue en estimant la probabilité d'occurrence de cette classe dans un grand corpus de texte. Le contenu de l'information (IC) d'un concept / d'une classe c est :

$$IC(c) = -logP(c)$$

où $P(c)$ est la probabilité de rencontrer une instance du concept c.

Resnik La mesure proposée par [Resnik, 1995] retourne simplement le contenu de l'information (IC) du plus bas ancêtre commun (LCS) de deux concepts donnés :

$$sim_{res}(c_1, c_2) = IC(LCS(c_1, c_2)) = -logP(LCS(c_1, c_2))$$

Lin La mesure proposée par [Lin, 1998] est une normalisation de celle de [Resnik, 1995]. La normalisation est faite en factorisant par le contenu de l'information (IC) des deux concepts.

$$sim_{lin}(c_1, c_2) = \frac{2*IC(LCSc_1,c_2)}{IC(c_1)+IC(c_2)}$$

Jiang and Conrath Enfin, une autre mesure proposée par [Jiang and Conrath, 1997], elle aussi basée sur celle de [Resnik, 1995], détermine la similarité comme suit :

$$sim_{jnc}(c_1, c_2) = \frac{1}{IC(c_1)+IC(c_2)-2IC(LCS(c_1,c_2))}$$

Exemple : A partir de la taxonomie de la figure 1.1, si nous considérons les concepts *vache*, *poulain* et *étalon*, sachant que le LCS de *vache* et *étalon* est *herbivore* et le LCS de *étalon* et *poulain* est *cheval*, et si nous supposons que $IC(vache) = IC(etalon) = 0,5$, $IC(poulain) = 0,6$, $IC(herbivore) = 0,3$ et $IC(cheval) = 0,4$, nous avons :
$sim_{res}(vache, etalon) = IC(LCS(vache, etalon)) = IC(herbivore) = 0,3$ et
$sim_{res}(etalon, poulain) = IC(LCS(etalon, poulain)) = IC(cheval) = 0,4$,
$sim_{lin}(vache, etalon) = \frac{2*IC(herbivore)}{IC(vache)+IC(etalon)} = 0,6$ et
$sim_{lin}(etalon, poulain) = \frac{2*IC(cheval)}{IC(etalon)+IC(poulain)} = 0,727$,
$sim_{jnc}(vache, etalon) = \frac{1}{IC(vache)+IC(etalon)-2IC(herbivore)} = 2,5$ et
$sim_{jnc}(etalon, poulain) = \frac{1}{IC(etalon)+IC(poulain)-2IC(cheval)} = 3,33$.

3.2.3 Approches statistiques (ou corpus-based)

Les mesures basées sur des corpus diffèrent des mesures présentées précédemment car elles ne nécessitent pas la compréhension du vocabulaire ou de la grammaire de la langue d'un texte. Parmi de telles mesures de similarité sémantique, nous présentons l'analyse sémantique latente (LSA) [Deerwester et al., 1990] où les co-occurrences de termes dans un corpus sont capturées au moyen d'une réduction de dimension réalisée par une décomposition en valeurs singulières (SVD) sur la matrice termes/documents représentant le corpus ; l'analyse sémantique explicite (ESA) [Gabrilovich and Markovitch, 2007] qui est une variation du modèle standard vectoriel où les dimensions du vecteur sont directement équivalentes à des concepts abstraits. D'autres mesures comme la distance normalisée de Google (Normalized Google Distance (NGD)) [Cilibrasi and Vitanyi, 2007] et le n^o de wikipédia (n^o of Wikipedia (noW)) [Wong et al., 2006] existent mais ne sont pas présentées en détail.

LSA / PLSA et LDA

[Deerwester et al., 1990] propose l'analyse sémantique latente (LSA), qui peut être utilisée pour déterminer la distance entre des mots ou entre des ensembles de mots. Contrairement aux diverses approches décrites précédemment où une matrice de co-occurrence mots/mots est créée, la première étape de la LSA consiste à créer des matrices mots/paragraphes, mots/documents ou mots/passages, où un passage est un groupe de mots. Une cellule pour un mot w et un passage p est, par exemple, rempli avec le nombre de fois qu'apparait w dans p. Ensuite, la dimension de cette matrice est réduite par l'application d'une décomposition en valeurs singulières (SVD), une technique de décomposition de matrice standard. Ce plus petit ensemble de dimensions représente un résumé (inconnu) de concepts. Puis la matrice originale mot/passage est recréée, mais cette fois à partir des dimensions réduites.

Lorsque vous utilisez la modélisation latente, les documents d'une collection sont modélisés comme une combinaison pondérée des thèmes latents d'un ensemble $Z = \{z_1, ..., z_{N_Z}\}$. Dans l'analyse sémantique latente probabiliste (PLSA) [Hofmann, 1999], chaque thème latent possède un modèle de langage probabiliste $P(w\|z)$ représentant la probabilité que le mot w puisse être généré par le thème z. Chaque document d_i de la collection de documents D est alors supposé avoir été généré par un mélange pondéré des modèles latents des thèmes. Si un document est modélisé par une collection de mots $C = \{c_1, ..., c_{N_V}\}$, le modèle génératif PLSA de C sachant d_i est :

$$P(C\|d_i) = \prod_{w \in V} (\sum_{z \in Z} P(w\|z) P(z\|d_i))^{c_w}$$

La distribution latente de Dirichlet (LDA) [Blei et al., 2003], quant à elle, est une généralisation de PLSA dans laquelle l'estimation ponctuelle de $P(z\|d_i)$ pour le document d_i dans PLSA est remplacée par une distribution probabiliste a priori de Dirichlet sur toutes les distributions possibles des thèmes latents au sein de Z.

Ces trois approches pâtissent de leur effet "boite noire".

ESA

L'analyse sémantique explicite (ESA) [Gabrilovich and Markovitch, 2007] est une représentation vectorielle de texte (mots isolés ou documents) qui utilise Wikipédia comme une base de connaissances. Plus précisément, dans l'ESA, un mot est représenté par un vecteur colonne de la matrice tf-idf du texte de l'article dans

Wikipédia et un document (chaîne de mots) est représenté comme le barycentre des vecteurs représentant ses mots.

L'ESA fait l'hypothèse que les articles de Wikipédia sont "orthogonaux". Toutefois, il a été démontré que l'ESA améliore également les performances des systèmes de recherche d'information quand elle est fondée non pas sur Wikipédia, mais sur le corpus Reuters, qui ne satisfait pas la propriété d'orthogonalité.

3.3 Outils

La plupart de ces mesures ont déjà été développées sur de nombreux supports. Cette section fera l'objet de recherches plus approfondies lorsqu'une décision aura été prise quant à la solution à adopter.

3.4 Comparatif

3.4.1 Performances

Nous tentons ici d'identifier les performances de chaque mesure, quelle soit sémantique ou syntaxique.

[Grefenstette, 2009] vante les mérites de l'approche basée sur les vecteurs sémantiques. D'après les auteurs, cette approche détecte aisément des documents similaires avec peu d'erreurs. [Bisson and Hussain, 2008] indique que le bi-clustering semble meilleur que le LSA et encore meilleur que la mesure basée sur le cosinus.

Bien que [Takale, 2007, Corley and Mihalcea, 2005, Mihalcea et al., 2006] assument que les méthodes sémantiques ont des performances similaires, [Budanitsky and Hirst, 2006] a tendance à les ordonner en considérant que la méthode de Jiang et Conrath a des performances supérieures aux autres, suivi par celle de Lin et celle de Leacock et Chodorow puis par celle de Resnik et [Jiang and Conrath, 1997] indique que les méthodes *node-based* semblent avoir de meilleurs résultats que les méthodes *edge-based*. Il faut également noter que [Corley and Mihalcea, 2005] indique que les mesures de Jiang et Conrath, de Leacock et Chodorow, de Lin, de Wu et Palmer et de Resnik sont meilleures que les approches vectorielles.

[Hazen, 2010] a montré que, dans certains cas, le LDA a des performances très décevantes par rapport aux approches vectorielles basées sur le tf-idf. Les résultats de [Mohler and Mihalcea, 2009] indiquent que les résultats obtenus avec les mesures basées sur la connaissance (knowledge-based) et celles basées sur un corpus (LSA et ESA) ont des performances comparables. L'avantage des approches basées sur un corpus par rapport à celles basées sur la connaissance réside dans leur indépendance de la langue et à leur facilité relative à créer des corpus spécifiques à un domaine contrairement à une taxonomie comme Wordnet.

Finalement, [Iosif and Potamianos, 2010] indique que les mesures de similarité de Jiang et Conrath et de Leacock et Chodorow ont des performances bien meilleures que celles des mesures basées sur le coefficient de Jaccard et l'indice de Dice.

3.4.2 Avantages et Inconvénients

Comparativement aux approches syntaxiques, certains inconvénients ont été palliés, d'autres subsistent. En effet, les problèmes liés à la négation et l'antinomie, aux rôles sémantiques inverses et à l'inconsistance logique ne sont toujours pas réglés.

De plus, il est à noter que les approches sémantiques basées sur les corpus ou la connaissance posent des problèmes de stockage et de complexité et sont souvent spécifiques à un domaine donné.

Chapitre 4

Synthèse

4.1 Tableau synthétique

Le tableau suivant récapitule toutes les méthodes présentées dans ce rapport.

Méthode	Approche syntaxique			Approche sémantique			
	Espace vectoriel	Edition	Termes communs	Espace vectoriel	Edge-based	Node-based	Corpus-based
Similarité Cosinus	X						
Coefficient de corrélation de Pearson	X						
Distance euclidienne	X						
Coefficient de Jaccard	X						
Distance de Levenshtein		X					
Indice de Dice			X				
Vecteurs sémantiques				X			
Bi-clustering				X			
Leacock et Chodorow					X		
Wu et Palmer					X		
Resnik						X	
Lin						X	
Jiang et Conrath						X	
LSA - PLSA - LDA							X
ESA							X

4.2 Décision

Tout au long de ce rapport, pour chaque méthode existante présentée, nous avons tenté de donner les avantages et les inconvénients. La décision finale de la méthode à développer doit être prise par le responsable du projet.

Ce document ne fait donc que présenter les méthodes existantes semblant être appropriées à notre compréhension du contexte.

Bibliographie

[Baake et al., 2006] Baake, M., Grimm, U., and Giegerich, R. (2006). Surprises in approximating levenshtein distances. *Journal of Theoretical Biology*, pages 279–282.

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Bavi et al., 2010] Bavi, V., Beirne, T., Bone, N., Mohr, J., and Neal, B. (2010). Comparison of document similarity metrics. Computer Science Department, Western Washington University, Information Retrieval, Winter 2010.

[Bisson and Hussain, 2008] Bisson, G. and Hussain, S. F. (2008). Chi-sim : A new similarity measure for the co-clustering task. In *ICMLA*, pages 211–217.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3 :993–1022.

[Budanitsky and Hirst, 2006] Budanitsky, A. and Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1) :13–47.

[Cilibrasi and Vitanyi, 2007] Cilibrasi, R. L. and Vitanyi, P. M. B. (2007). The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3) :370–383.

[Corley and Mihalcea, 2005] Corley, C. and Mihalcea, R. (2005). Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6) :391–407.

[Fellbaum, 1998] Fellbaum, C., editor (1998). *WordNet : An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, illustrated edition edition.

[Gabrilovich and Markovitch, 2007] Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611.

[Grefenstette, 2009] Grefenstette, E. (2009). Analysing document similarity measures. Master's thesis, University of Oxford.

[Hazen, 2010] Hazen, T. J. (2010). Direct and latent modeling techniques for computing spoken document similarity. In *SLT*, pages 366–371.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. ACM.

[Huang, 2008] Huang, A. (2008). Similarity Measures for Text Document Clustering. In Holland, J., Nicholas, A., and Brignoli, D., editors, *New Zealand Computer Science Research Student Conference*, pages 49–56.

[Iosif and Potamianos, 2010] Iosif, E. and Potamianos, A. (2010). Unsupervised semantic similarity computation between terms using web documents. *IEEE Transactions on Knowledge and Data Engineering*, 22(11) :1637–1647.

[Jaccard, 1901] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37 :547–579.

[Jiang and Conrath, 1997] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008.

[Leacock and Chodorow, 1998] Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In Fellfaum, C., editor, *MIT Press*, pages 265–283, Cambridge, Massachusetts.

[Lesk, 1986] Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries : how to tell a pine cone from an ice cream cone. In *SIGDOC '86 : Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, New York, NY, USA. ACM Press.

[Levenshtein, 1966] Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8) :707–710.

[Lin, 1998] Lin, D. (1998). An information-theoretic definition of similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann.

[Mihalcea et al., 2006] Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *IN AAAIŠ06*, pages 775–780.

[Mohammad and Hirst, 2012] Mohammad, S. and Hirst, G. (2012). Distributional measures of semantic distance : A survey. *CoRR*, abs/1203.1858.

[Mohler and Mihalcea, 2009] Mohler, M. and Mihalcea, R. (2009). Text-to-text semantic similarity for automatic short answer grading. In *In Proc. of EACL*.

[Navarro, 2001] Navarro, G. (2001). A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1) :31–88.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and jing Zhu, W. (2002). Bleu : a method for automatic evaluation of machine translation. pages 311–318.

[Patwardhan et al., 2003] Patwardhan, S., Banerjee, S., and Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In *CICLing*, pages 241–257.

[Porter, 1980] Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14(3) :130–137.

[Resnik, 1995] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI 1995*, pages 448–453.

[Rocchio, 1971] Rocchio, J. (1971). *Relevance Feedback in Information Retrieval*, pages 313–323.

[Sagot and Fišer, 2008] Sagot, B. and Fišer, D. (2008). Building a free French wordnet from multilingual resources. In *OntoLex*, Marrakech, Maroc.

[Salton, 1997] Salton, G. (1997). Automatic text structuring and summarization. *Information Processing & Management*, 33(2) :193–207.

[Strehl et al., 2000] Strehl, A., Ghosh, J., and Mooney, R. (2000). Impact of Similarity Measures on Web-page Clustering. In *Proceedings of the 17th National Conference on Artificial Intelligence : Workshop of Artificial Intelligence for Web Search (AAAI 2000), 30-31 July 2000, Austin, Texas, USA*, pages 58–64. AAAI.

[Takale, 2007] Takale, S. A. (2007). Measuring semantic similarity between words using web documents.

[Wong et al., 2006] Wong, W., Liu, W., and Bennamoun, M. (2006). Featureless similarities for terms clustering using tree-traversing ants. In *Proceedings of the 2006 international symposium on Practical cognitive agents and robots*, PCAR '06, pages 177–191, New York, NY, USA. ACM.

[Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.