Acquisition automatique de la morphologie d'une langue

tour d'horizon

Aurélien Moreau

3 avril 2003

Prélude (bases linguistiques)

Définitions :

La morphologie est l'étude de la structure/composition d'un mot.

Le morphème est l'unité de base, plus petite division d'un mot porteuse de sense.

Les flexions, dérivations sont des modifications de mot (grâce à des morphèmes).

Exemple :

anticonstitutionnellement $\rightarrow$

anti(flexion)-constitu[er](racine)-tion(dérivation)-nelle-ment(dérivation)

Plan

1. Introduction (domaine d'application et quelques résultats de complexité).
2. Les différentes approches :
   – Apprentissage à base d'exemples.
   – La morphologie à deux niveaux et l'apprentissage de ses règles.
   – Connexionnisme et ILP (inductive logic programming), les autres approches.
3. Apprentissage sur fond de sémantique.
4. Conclusion.

Introduction

Les domaines d'applications sont plus ou moins variés :

- Recherche d'information.
- Lemmatisation (revenir à la racine).
- Traduction ?

On peut voir l'analyse morphologique comme un "super" tagger.

Idée sur la complexité $\rightarrow$ NP-difficile pour l'acquisition morpho-phonologique[7], Manning explique la difficulté de la segmentation[5].

Apprentissage à base d'exemples

MBMA(Memory-based morphological analysis) par A. Van den Bosch et W. Daelemans[3].
Utilisation d'une machine d'apprentissage par l'exemple IB1-ig (information gain).
Les exemples (langue allemande)sont tirés de CELEX[a] (une base de donnée lexicale).
On ramène le problème d'analyse morphologique à un problème de classification des lettres. Les points importants :

– Créer les instances de notre base d'exemples.
– Classer les nouveaux mots.
– Les résultats.

[a] http ://europa.eu.int/celex/

Instanciation

Technique de fenêtrage pour l'instanciation focalisé sur une lettre.
Utilisation des 247415 mots de CELEX entièrement analysés pour créer les 2727462 instances de la base d'exemple

numéro d'instance	contexte gauche	lettre cible	contexte droit	classes
1		i	nvari	A_*A
2	i	n	varia	0
3	in	v	ariab	V+Der
4	inv	a	riabl	0
5	inva	r	iable	0
6	invar	i	ablem	0
7	nvari	a	bleme	A_V*
8	varia	b	lemen	0
9	ariab	l	ement	0
10	riabl	e	ment	0
11	iable	m	ent	B_A*
12	ablem	e	nt	0
13	bleme	n	t	0
14	lemen	t		0

Classification

Utilisation d'une distance (IB1-ig) :

$$\Delta(X,Y) = \sum_{i=1}^{n} W(f_i)\delta(x_i, y_i) \tag{1}$$

pour comparer à chaque élément de la base d'exemple. La classe de l'instance en mémoire Y qui minimise $\Delta(X,Y)$ est prise pour X l'instance étudiée.

Résultats

Deux types de résultats sont offerts, sur des instances de test (10-fold cross-validation) :

granularité	exemple	instances	mots
baseline	tout mis à l'instance 0	72,5%	
complète	$[\text{in}]_{A\_*A}[\text{varier}]_{V+Der}[\text{able}]_{A\_V*}[\text{ment}]_{B\_A*}$	95,88%	64,63%
dériv/inflect	$[in]_{infl}[varier]_{deriv}[able]_{deriv}[ment]_{deriv}$	98,83%	89,62%
segmentation	$[in][varier][able][ment]$	98,97%	90,69%

granularité	précision	rappel
complète	84,33%	83,76%
dériv/inflect	94,72%	94,07%
segmentation	94,83%	94,18%

ainsi qu'une estimation de résultats sur un vrai texte :

granularité	instances	mots
complète	98,4%	84,2%
dériv/inflect	99,6%	96,7%
segmentation	99,6%	96,7%

granularité	précision	rappel
complète	93,6%	93,2%
dériv/inflect	98,5%	98,0%
segmentation	98,5%	97,9%

Morphologie à deux niveaux

Créée en partie pour formaliser les transformations qui s'opèrent lors de la concaténation de morphèmes (permet aussi une analyse morphologique voir PC-Kimmo). On a un niveau **lexical** ainsi qu'un niveau de **surface** :

niveau lexical :	v	e	u	f	+	e
niveau de surface :	v	e	u	v	#	e

On définit alors des règles décrivant les changements à effectuer selon le contexte : f :v $\Leftrightarrow$ V :V - <+ :#,e :e>
Trois opérateurs sont utilisés répondant au 2 questions qui suivent :

1. Est-ce que E est le seul environnement ou on a L :S.
2. L doit-il toujours être changé en S dans l'environement E.

opérateurs	réponse à Q1	réponse à Q2
$\Rightarrow$	OUI	NON
$\Leftarrow$	NON	OUI
$\Leftrightarrow$	OUI	OUI

Génération automatique de ces règles

– Travail de Theron et Cloete[10].
– Se base sur une liste de paires **{lemme,cible}**.
– La technique segmente les mots puis induit des règles optimales.
– Utile pour les langues qui ne disposent pas de ces règles.

Segmentation

1. Se fait par le calcul de distance d'édition entre la cible et le lemme. Calcul de la séquence de distance minimale (coût égaux, pas de remplacement de lettre). Si plusieurs on garde celle qui à le schéma INSERER...,EFFACER,PASCHG...,EFFACER,INSERER... Exemple

   (happy,unhappier) :

u	n	h	a	p	p	0	i	e	r
0	0	h	a	p	p	y	0	0	0
I	I	P	P	P	P	E	I	I	I

2. Puis fusion de toutes les séquences dans un automate minimale AFSA (acyclic finite state automaton) qui ne reconnait que les séquences trouvées avec maintient des comptes des arrêtes (les opérations).

3. Les séquences interressantes sont celles correspondantes aux INSERER avec des comptes similaires. Segmentation en parcourant le DAG (côté destination) avec ajout du caractère '+'. Exemple

Induction des règles(1)

1. Obtenir la surface par distance d'édition (ajout de la substitution avec un coût identique). On obtient :

lexical	u	n	+	h	a	p	p	y	+	e	r
surface	u	n	0	h	a	p	p	i	0	e	r
opération	P	P	I	P	P	P	P	S	I	P	P

2. Repérer le changement (substitution pertinante, on appelle paire marquée par la suite) $\Rightarrow$ devient la base pour la nouvelle règle.
3. Il reste à trouver le contexte optimal et l'opérateur correspondant.

Induction des règles(2)

Solution : un AFSA (encore) sur les contextes mixtes de toutes les règles potentielles. Puis recherche de chemin répondant aux attentes des opérateurs, pour ceci on traduit les questions par des propriétés sur le DAG :

Question 1 : Si tous les chemins de la racine à L :S passe par une arrête e_1 alors OUI pour E contexte jusqu'a e_1.

Question 2 : Si pour tous les arrêtes L :S_i qui partagent une arrête e_2 les S_i sont égaux alors OUI pour E contexte contexte jusqu'a e_2.

Marquer tous les chemins en parcourant le DAG puis faire sont choix (préférence pour l'opérateur $\Leftrightarrow$).

Résultats(1)

Exmples de résultats qu'on peut obtenir, basé sur le travail de Antworth 1990[1] (PC-KIMMO), seulement 21 paires **{lemme,cible}**.

Cible		Préfixe+	Lemme	+Suffixe
bigger	=		big	+er
biggest	=		big	+est
unclear	=	un+	clear	
unclearly	=	un+	clear	+ly
unhappy	=	un+	happy	
unhappier	=	un+	happy	+er
unhappiest	=	un+	happy	+est
unhappily	=	un+	happy	+ly
unreal	=	un+	real	
cooler	=		cool	+er
coolest	=		cool	+est
coolly	=		cool	+ly
clearer	=		clear	+er
clearest	=		clear	+est
clearly	=		clear	+ly
redder	=		red	+er
reddest	=		red	+est
really	=		real	+ly
happier	=		happy	+er
happiest	=		happy	+est
happily	=		happy	+ly

Résultats(2)

Les règles généralisés sont :

1. 0 :d ⇔ d :d _ + :0
2. 0 :g ⇔ g :g _ + :0
3. y :i ⇔ _ + :0

Un autre test sur du Xhosa existe, montre des règles plus compliquées.

Ikhanto Yesithathu

Ziporhondini, 'ntezisabe kwiakile yezidumbu, igazi lilihlwili kumabunzi enu - gobho umngxum' omnye kubuchopho benu - nihamba kumabhodlo aqhumayo eemfazwe zethu zobukhaya : Aneyiswanga. Nazalwa Negeminyaka yengcwakaha, nalinda elenu ixesha : nithe nisolula iflegi ebomvu yezindlu namakhaya atshayo, niphakamisel' iinqwelo zomkhosi amanqindi enu aze, nasuka natyotyozwa ngamavili eeratele. Ukuba cekecekana komzimba womntu uqubisene nentsimbi : kodwa nihambile ubuso buze, nafunza koombayi -mbayi. Kuba, umntu naxa enokutshatyalaliswa, uyoyikwa bugoxo bezixhobo obenziwe ngeentsimbi ziimfama. Kaloku akukho nto yomelele ngaphezu kwale nyam' ibolayo, xa ihamba igwaba isinga kwizixhobo zokufa. Kwaye 'khe wahamba, ungofileyo, kodwa uyahamba : ithi ke imizimba yenu edlavulweyo iqhambukise unkwantyo phakathi kwabalawuli.

Les autres approches

Toutes ces approches travaillent la plupart du temp sur l'apprentissage des formes passés des verbes anglais.

- Le connectionnisme[8]. Très controversée[4].
- Programmation logique, ILP(induction logic programming)[6]. Interressant par ces résultats.
- PHMM[2] (Pair Hidden Markov Model) qui provient de la bioinformatique. Permet d'après l'auteur d'acquérir de vraies morphologies de langues à partir de sources non annotées.

Un peu de sémantique

- S'interresser au mot de façon purement "syntaxique" pose certain problème comme ***ally*** réduit en ***all*** après lemmatisation ou encore ***rating*** réduit en ***rat*** à la place de ***rate***.
- Schone et Jurafsky [9] proposent de s'appuyer sur LSA (Latent Semantic Analysis) pour apparenter des mot ayant la même racine. Etapes de leur méthode :
  1. Identifier les candidats afixes (dépos dans un trie).
  2. Identifier les paires de mots qui peuvent être des variants (toujours grâce au trie).
  3. Trouver les vecteurs sémantiques des mots (méthode des valeures singulières).
  4. Filtrer les paires qui n'ont pas de correspondance sémantique (cosinus normalisé).

Bibliography

Références

[1] Evan L. Antworth. Pc-kimmo : a two-level processor for morphological analysis. occasional publications in academic computing no. 16. dallas, tx : Summer institute of linguistics., 1990.

[2] Alexander Clark. Learning morphology with pair hidden markov models. In *ACL (Companion Volume)*, pages 55–60, 2001.

[3] A. Van den Bosch and W. Daelemans. Memory-based morphological analysis, 1999.

[4] Charles X. Ling. Learning the past tense of english verbs : The symbolic pattern associator vs. connectionist models. *Journal of Artificial Intelligence Research*, 1 :209–229, 1994.

[5] Christopher D. Manning. The segmentation problem in morphology learning. In Jill Burstein and Claudia Leacock, editors, *Proceedings of the Joint Conference on New Methods in Language Processing and Computational Language Learning*, pages 299–305. Association for Computational Linguistics, Somerset, New Jersey, 1998.

[6] R.J. Mooney and M.E. Califf. Induction of first-order decision lists : Results on learning the past tense of English verbs. In L. De Raedt, editor, *Proceedings of the 5th International Workshop on Inductive Logic Programming*, pages 145–146. Department of Computer Science, Katholieke Universiteit Leuven, 1995.

[7] Eric Sven Ristad. Complexity of morpheme acquisition.

[8] D. Rumelhart and J. McClelland. On learning the past tenses of english verbs, 1986.

[9] Patrick Schone and Daniel Jurafsky. Knowledge-free induction of morphology using latent semantic analysis. In Claire Cardie, Walter Daelemans, Claire Nédellec, and Erik Tjong Kim Sang, editors, *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, 2000*, pages 67–72. Association for Computational Linguistics, Somerset, New Jersey, 2000.

[10] Pieter Theron and Ian Cloete. Automatic acquisition of two-level morphological rules, 1997.