ETIQUETAGE GRAMMATICAL AUTOMATISE DE TEXTES

Baptiste LAGARDE

Octobre 2006

Table des matières

I Introduction **3**

II Présentation de l'application **4**

1 Contexte **4**

2 Objectifs **4**

3 Principe **5**
3.1 Aprentissage 5
3.2 Etiquetage 5

III Exemple de fonctionnement **5**

4 Point de départ **5**
4.1 Jeu d'étiquettes 5
4.2 Corpus 6
4.3 Texte à étiqueter 6

5 Apprentissage **6**
5.1 Théorie 6
5.2 Pratique 7

6 Etiquetage **7**
6.1 Phrase à étiqueter 7
6.2 Etiquetage des mots non ambigus 7
6.3 Etiquetage des mots ambigus 8

IV Conclusions **9**

7 Limites du programme **9**

8 Domaines d'application **9**

Première partie
Introduction

Rendre chaque jour plus ergonomiques les outils et les machines que nous concevons est capital. Pour cela, un travail important est à fournir sur l'interface homme/machine. La manière idéale de commander n'importe quel système automatisé serait le langage naturel. N'importe quel utilisateur aurait ainsi la possibilité d'effectuer le pilotage sans connaissances supplémentaires en informatique.
Mais les langages naturels (les langages utilisés par les hommes entre eux) ont la particularité de présenter de l' ambiguité, contrairement aux langages formels (langages de programmation, par exemple). L'ambiguité se situe à plusieurs niveaux du langage :

- Pour la langue parlée, elle peut être au niveau des sons constituant le signal de parole (on demande à son interlocuteur de répéter), ou encore au niveau du sens (on lui demande alors de préciser le contexte).
- Pour la langue écrite, l'ambiguité peut se trouver au niveau semantique, mais elle existe aussi dans le manque de clarté de l'écriture, ou encore dans la présence d'homographes. C'est ce dernier point auquel nous allons nous intéresser.

Afin d'effectuer n'importe quel traitement sur des informations que l'on fournit à une machine, celle-ci doit pouvoir lever d'elle-même l'ambiguité qu'elles contiennent. C'est en analysant le contexte (comme le ferait un interlocuteur humain) qu'elle y parvient. Des procédés pour lever l'ambiguité sur le langage naturel existent déjà à tous les niveaux, et sont largement répandus dans les logiciels de reconnaissance vocale, de traduction automatique, d'applications dialogantes, etc.

Ce document présente donc un algorithme d'apprentissage à partir d'un corpus, implémenté sous Matlab, et permettant à un ordinateur d'associer automatiquement à tous les mots d'un texte saisi une étiquette donnant sa nature grammaticale.

Deuxième partie

Présentation de l'application

1 Contexte

Afin qu'une machine réalise ce que lui demande son opérateur en langage naturel, il faut qu'elle "comprenne" ce qu'il lui demande. Pour un ordinateur comme pour un humain, l'interprétation du sens (la sémantique) d'une phrase se fait dans un ordre précis correspondant aux différentes couches du langage dans son modèle vertical (Fig.1). L'ambiguité est présente dans chacune de ces couches, donc lors de la conception d'un analyseur travaillant sur l'une d'entre elles, il faut pouvoir propager les incertitudes au niveau suivant. Nous travaillerons donc avec des probabilités, qui peuvent être réévaluées à chaque niveau.

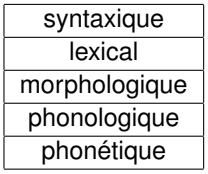

Fig. 1 – Modèle vertical du langage naturel

Pour illustrer cela, imaginons un analyseur lexical qui déclare que la liste de phonèmes qu'il a récupéré en sortie d'une application de reconnaissance vocale correspond à "*il part*" avec une probabilité de 0,3 et à "*île port*" avec une probabilité de 0,7 (à cause de l'accent du locuteur, par exemple). L'analyseur syntaxique qui suit réévaluera ces deux probabilités grâce à ses propres critères (syntaxiques), et réfutera donc l'interprétation "île port" dénuée de sens, qui sortait pourtant favorie à l'étape précédente. Bien entendu, propager ainsi les ambiguités demande d'énormes ressources calculatoires, mais c'est nécessaire car chaque couche du langage participe à la sémantique.

Le programme présenté dans ce document se situe au niveau syntaxique : il s'appuie donc sur l'ordre des mots pour en dégager le sens.

2 Objectifs

Nous cherchons à concevoir une application qui, étant donné un texte, retourne une liste d'étiquettes correspondant à la nature grammaticale de chacun de ses mots, et les probabilités associées. La phrase "*Marie le juge*" aura donc à être interprétée comme :

Mots	Marie	le	juge
Nature	NPfs	CL3ms [1]	VP1p
Probabilité associée	0.3	0.8	0.6

ou encore, selon le contexte :

Mots	Marie	le	juge
Nature	VY2s [2]	Dms	NCms
Probabilité associée	0.6	0.2	0.4

[1] Clitique 3ème personne masculin singulier
[2] Verbe impératif 2ème personne du singulier

3 Principe

Le programme fonctionne en deux temps :

1. Une phase d'apprentissage
2. Une phase d'étiquetage

3.1 Aprentissage

Dans ce programme, nous limiterons l'étude à des bigrammes-mots, en n'appelant "contexte d'un mot" que les mots immédiatement adjacents à celui-ci.

Afin que l'ordinateur puisse repérer le contexte dans lequel se trouve un mot, il faut lui faire faire un apprentissage. L'idée de cet apprentissage est de lui soumettre un corpus manuellement étiqueté (Fig.2), pour lequel il comptera le nombre d'occurences de chaque graphème (mot écrit avec une orthographe donnée) muni de chaque étiquette, encadré par chaque couple d'étiquettes.

```
Maintenant :ADV
       ,
  tout :PROms
   est :VP3s
     à :P
  refaire :VW
       .
```

Fig. 2 – extrait de corpus étiqueté

Il doit pour cela bien entendu disposer du jeu d'étiquettes grâce auquel a été étiqueté le corpus. Finalement, l'ordinateur convertira ces nombres d'occurences en probabilités, et ce sont celles-ci qui serviront à déterminer la nature la plus probable de chaque mot du texte à étiqueter.

3.2 Etiquetage

Le programme doit ensuite attribuer une nature grammaticale (et la probabilité associée) à chaque mot du texte fourni. Cet étiquetage se déroule en deux temps :

1. On repère les mots qui ne présentent aucune ambiguité.
2. On détermine grâce aux mots adjacents la nature la plus probable des autres mots.

Troisième partie
Exemple de fonctionnement

4 Point de départ

Dans cette partie, nous allons voir concrètement comment le programme se constitue une base de connaissances à partir du corpus et du jeu d'étiquettes associé.

4.1 Jeu d'étiquettes

Le nombre N_T d'étiquettes n'est pas universel pour une langue donnée. Pour la clarté de l'exposé, nous utilisons un jeu simplifié (Fig.3), qui confond par exemple “Déterminant masculin singulier” et “Déterminant féminin pluriel” en “Déterminant”.

PONCT	AC	AI	A	ADVE	ADV	CC
CS	CLS	CLO	CLR	CL	DE	DP
D	ET	I	NC	NP	P	PREF
PROI	PROR	PRO	VC	VF	VG	VI
VJ	VK	VP	VS	VT	VW	VY

FIG. 3 – Jeu d'étiquettes simplifié

4.2 Corpus

Pour la phase d'apprentissage, nous utilisons un "mini-corpus" (FIG.4) qui contient de l'ambiguité. Ce corpus aura été étiqueté manuellement, ou bien de manière semi-automatique (par un autre programme, puis corrigé manuellement).

```
Pignon : Ah bon, il n'a pas de prénom?
Pierre : Je viens de vous le dire : Juste Leblanc. Leblanc, c'est son nom, et
c'est Juste son prénom.
Pignon : Aah...?
Pierre : Monsieur Pignon : votre prénom, à vous, c'est François. ... C'est juste?
Pignon : Oui...
Pierre : Eh bien, lui, c'est pareil, ....c'est Juste.
```

FIG. 4 – Corpus utilisé (l'étiquetage n'est pas représenté)

Le mot "juste" existe en français avec les natures d'adjectif, d'adverbe, de nom commun, et de nom propre. Il présente donc une ambiguité pour notre étiqueteur automatique. Dans ce texte, il n'apparaît qu'avec les natures de nom propre et d'adjectif attribut, mais c'est suffisant pour expliquer le fonctionnement du programme.

4.3 Texte à étiqueter

Celui-ci ne joue un rôle que lors de la deuxième phase, qui est l'étiquetage proprement dit. Le texte ne doit doit comporter que des mots qui auront été vus lors de la phase d'apprentissage, c'est à dire des mots du corpus.

5 Apprentissage

5.1 Théorie

Tout d'abord, un **lexique** du corpus est constitué. Il s'agit d'une liste où chaque mot du corpus ne figure qu'une seule fois.
Puis, pour chaque mot m_i du lexique, on compte son nombre d'apparitions dans le corpus avec l'étiquette t_j **et** suivi de l'étiquette t_k. Grâce à la formule de Bayes[3], en divisant toutes ces valeurs par le nombre de fois où l'étiquette du mot suivant est t_k, on obtient la matrice des probabilités conditionnelles :

$$P_{sachantLeMotSuivant}(i) = \begin{bmatrix} p(t(m_i) = t_1 / t(m_{suivant}) = t_1) & \cdots & p(t(m_i) = t_{N_T} / t(m_{suivant}) = t_1) \\ \vdots & \ddots & \vdots \\ p(t(m_i) = t_1 / t(m_{suivant}) = t_{N_T}) & \cdots & p(t(m_i) = t_{N_T} / t(m_{suivant}) = t_{N_T}) \end{bmatrix}$$

[3] $p(A/B) = \frac{p(A \bigcap B)}{p(B)}$

5.2 Pratique

Dans le cas du mot “juste” on obtient la matrice nulle, à l’exception de :

$$p(t(“juste”) = NP/t(m_{suivant}) = NP) = 1$$
$$p(t(“juste”) = NP/t(m_{suivant}) = D) = 1$$
$$p(t(“juste”) = NP/t(m_{suivant}) = PONCT) = 0,5$$
$$p(t(“juste”) = A/t(m_{suivant}) = PONCT) = 0,5$$

Explication : Dans le corpus,

- “Juste” en tant que nom propre n’est suivi d’un nom propre que dans “Juste Leblanc”
- “Juste” en tant que nom propre n’est suivi d’un déterminant que dans “Juste son”
- “Juste” est suivi d’un signe de ponctuation dans “juste ?” et dans “Juste.” Donc sachant qu’il est suivi d’une marque de ponctuation, il a 50% de chances d’être un adjectif, et 50% de chances d’être un nom propre.

Bien entendu, ces statistiques ne peuvent avoir de valeur que pour des corpus très grands, mais cet exemple sert uniquement à montrer le fonctionnement du programme.
Pour le fonctionnement de l’algorithme d’étiquetage des mots ambigus, on construit aussi les matrices prenant en compte le mot précédent.
On pourrait aussi réaliser des apprentissages plus complexes en prenant en compte les deux mots adjacents, les trois mots adjacents, et ainsi de suite. On manipulerait alors des matrices de plus grandes dimensions, dont la constitution prendra plus de temps. Soulignons au passage l’importance d’optimiser l’implémentation de l’algorithme. En effet, pour un corpus de 50 000 mots comportant un lexique de 9288 mots, et des matrices uniquement relatives aux mots adjacents, un calcul non optimisé peut prendre 4h sous Matlab.

```
>> apprentissage('corpora/corpusMondeCorrected.iso', 'ecriture.txt')

longueur_lexique =

        9288

Elapsed time is 163.608098 seconds.
Elapsed time is 12449.199571 seconds.
>>
```

6 Etiquetage

6.1 Phrase à étiqueter

Choisissons une phrase à étiqueter : `Son prénom est Juste, et son nom Leblanc.`
Cette phrase ne doit comporter que des mots qui se trouvent dans le corpus.

6.2 Etiquetage des mots non ambigus

Celui-ci se fait en repérant les mots qui n’apparaissent dans le corpus qu’avec une seule nature grammaticale. Comme le seul mot à présenter une ambiguité dans la phrase précédente est “juste”, ce premier étiquetage donne :

Son	prénom	est	Juste	,	et	son	nom	Leblanc	.
D	NC	VP	?	CC	D	PONCT	NC	NP	PONCT
1	1	1	0	1	1	1	1	1	1

6.3 Etiquetage des mots ambigus

L'idée est de s'appuyer sur l'étiquetage précédent, et de travailler itérativement sur les mots adjacents à ceux dont on a des informations sur la nature grammaticale. Dans la phrase que nous utilisons en exemple, l'étiqueteur de mots ambigus ne regardera donc que le mot "juste", puisque c'est le seul pour lequel la probabilité associée est différente de 1.
Voici la théorie de la méthode : on fait deux apprentissages (l'un prenant en compte le mot précédent et l'autre le mot suivant). Si i est l'indice du mot dans la phrase que l'on souhaite étiqueter, et j et k les indices d'étiquettes appartenant au jeu choisi, on peut nommer A et B les évènements suivants :

$$A = \text{“}t(m_i) = t''_j$$
$$B = \text{“}t(m_{i+1}) = t''_k$$

Par ailleurs, écrire deux fois la formule de Bayes pour des évènements A et B quelconques nous donne :

$$p(A) = \frac{p(A/B)\cdot p(B)}{p(B/A)}$$

Voilà donc un moyen d'évaluer $p(A)$. Chaque terme du membre de droite est en effet connu :

- $p(A/B)$ est évalué dans la matrice des étiquettes sachant celle du mot suivant,
- $p(B/A)$ est évalué dans la matrice des étiquettes sachant celle du mot précédent,
- $p(B)$ est connu par hypothèse (on étiquette un mot adjacent à ceux dont on sait quelque chose de la nature grammaticale)

Pour chaque étiquette éventuelle pour le mot considéré, on peut donc évaluer deux valeurs de $p(A)$ associées à des étiquettes : une basée sur le mot précédent, et une sur le mot suivant. C'est la plus élevée des deux qui détermine l'étiquette qui sera choisie.
Terminons par l'application de ce que nous venons de voir à l'étiquetage du mot "juste" dans la phrase qu'on s'est donné. On a :

$$p(t(\text{“}juste'') = NP/t(m_{prec}) = VP) = \frac{2}{3}$$
$$p(t(\text{“}juste'') = A/t(m_{prec}) = VP) = \frac{1}{3}$$
$$p(t(\text{“}est'') = VP/t(m_{suiv}) = NP) = 1$$
$$p(t(\text{“}est'') = VP/t(m_{suiv}) = A) = 1$$

d'une part, et

$$p(t(\text{“}juste'') = NP/t(m_{suiv}) = PONCT) = \frac{1}{2}$$
$$p(t(\text{“}juste'') = A/t(m_{suiv}) = PONCT) = \frac{1}{2}$$
$$p(t(\text{“},'') = PONCT/t(m_{prec}) = NP) = 1$$
$$p(t(\text{“},'') = PONCT/t(m_{prec}) = A) = 1$$

D'après le bigramme "est Juste", le mot "Juste" est donc un nom propre avec la probabilité $\frac{2/3}{1}\cdot 1 \simeq 0,67$, et c'est un attribut avec la probabilité $\frac{1/3}{1}\cdot 1 \simeq 0,33$. D'après le bigramme "Juste," ("Juste VIRGULE"), "Juste" a autant de chances d'être un adjectif qu'un nom propre, avec une probabilité de $\frac{1/2}{1}\cdot 1 \simeq 0,5$. L'étiquette la plus probable aura donc été fournie par le bigramme "est Juste", et c'est un nom propre avec une probabilité de $\frac{2}{3}$.

Son	prénom	est	Juste	,	et	son	nom	Leblanc	.
D	NC	VP	NP	CC	D	PONCT	NC	NP	PONCT
1	1	1	0,67	1	1	1	1	1	1

Si nous avions eu une suite de plusieurs mots ambigus, nous aurions réitéré la phase d'étiquetage jusqu'à ce que chacun en ait une. En cas d'égalité de scores pour deux étiquettes ou plus, l'une d'entre elles aurait été choisie au hasard. On peut dans ce cas imaginer une sophistication du programme qui parcourrait l'arbre des possibilités d'étiquetage. On pourrait alors choisir de sélectionnner l'étiquetage qui maximise la somme des probabilités associées aux étiquettes des mots.

Quatrième partie
Conclusions

7 Limites du programme

Le programme présenté possède des défauts. Tout d'abord, il ne travaille "que" sur des bigrammes-mots : ceci constitue à la fois un avantage et un inconvénient : il est capable de traiter tous les cas de bigrammes qu'il aura rencontré lors de l'apprentissage, ce qui fait qu'il ne sera pas "dérouté" par l'apparition d'un trigramme inédit dans le texte à étiqueter. Un programme ayant réalisé son apprentissage par trigrammes l'aurait été car la probabilité associée à ce trigramme aurait été 0. Par contre, il aurait été beaucoup plus "sûr de lui". On peut donc conclure qu'un apprentissage par n-grammes est préférable dans le cas où le corpus est grand. Un inconvénient d'ordre plus général est que cet algorithme ne réalise l'étiquetage qu'a partir de critères syntaxiques, alors que d'autres informations de contexte (lexicales, par exemple) pourraient aider à lever les ambiguités. C'est là une manifestation de la contribution à la sémantique de chacune des couches du modèle vertical du langage. Enfin, ce programme travaille sur des corpus étiquetés de très grande taille. Cela pose un problème car l'étiquetage manuel est fastidieux donc coûteux, et les ressources en corpus étiquetés (bien qu'étendues) sont loin d'être illimitées. Cela limite donc sérieusement l'exploitation d'un tel programme.

8 Domaines d'application

Parmi les applications directes auxquelles nous pouvons penser pour cet étiqueteur de textes, figurent les logiciels de traduction automatique et les correcteurs orthographiques, mais aussi tous les systèmes présentant une interface homme/machine. On peut citer par exemple les applications de dialogue automatique, mais cela va bien plus loin. L'un des traits de caractère propres à l'Homme est l'humour, qu'il est très difficile de faire reproduire par une Intelligence Artificielle. Or, de nombreux jeux de mots sont fondés sur l'existence de plusieurs natures différentes pour un même mot. Si produire un moteur humoristique paraît ambitieux, on peut par contre raisonnablement envisager la faisabilité d'une application qui "comprenne" ce trait de caractère humain.