Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging

Eric Brill

Introduction

Un effort a récemment été entrepris pour créer des systèmes de traduction automatique dans lesquels les informations linguistiques nécessaires à la traduction sont automatiquement extraites de corpus alignés.

Les méthodes basées sur des corpus réussissent souvent tout en ignorant les véritables complexités du langage, en misant sur le fait que des phénomènes linguistiques complexes peuvent souvent être observés indirectement par le biais de simples épiphénomènes.

Exemple : *He will race/VERB the car.*
He will not race/VERB the car.
When will the race/NOUN end?

Approche : Transformation-based error-driven

Utile : étiquetages morphosyntaxique, analyse syntaxique et encore le technique lettre-son utilisé pour construire des réseaux de la reconnaissance de parole et les autres

Transformation-Based Error-Driven Learning

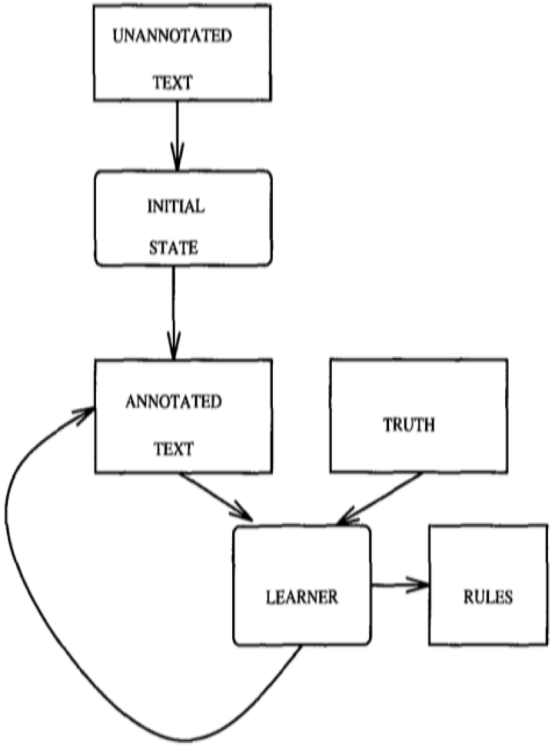
Ce système est basé sur la recherche et la correction d'erreurs. Lors de la période d'entraînement et à partir d'un corpus d'apprentissage étiqueté manuellement, le système reconnaît lui-même ses faiblesses et les corrige en construisant une base de règles.

- Deux types de règles sont utilisés dans l'étiqueteur d'Eric Brill:
- règles lexicales : ce type de règles permet de définir l'étiquette du mot en se basant sur ses propriétés lexicales.
- règles contextuelles : ce type de règles permet d'affiner l'étiquetage, c'est-à-dire de revenir sur les étiquettes précédemment affectées et de les corriger en examinant le contexte local.

Une fois entraîné, l'étiquetage se fait en deux étapes:

durant la première, chaque mot du texte reçoit l'étiquette la plus probable dans le contexte considéré, soit par consultation du lexique où le mot est connu, soit par application des règles lexicales si le mot est inconnu au lexique.

Pendant la seconde étape, le système revient sur ces premières affectations, examine le contexte local et corrige éventuellement les étiquettes précédemment affectées.



Exemple d'une transformation

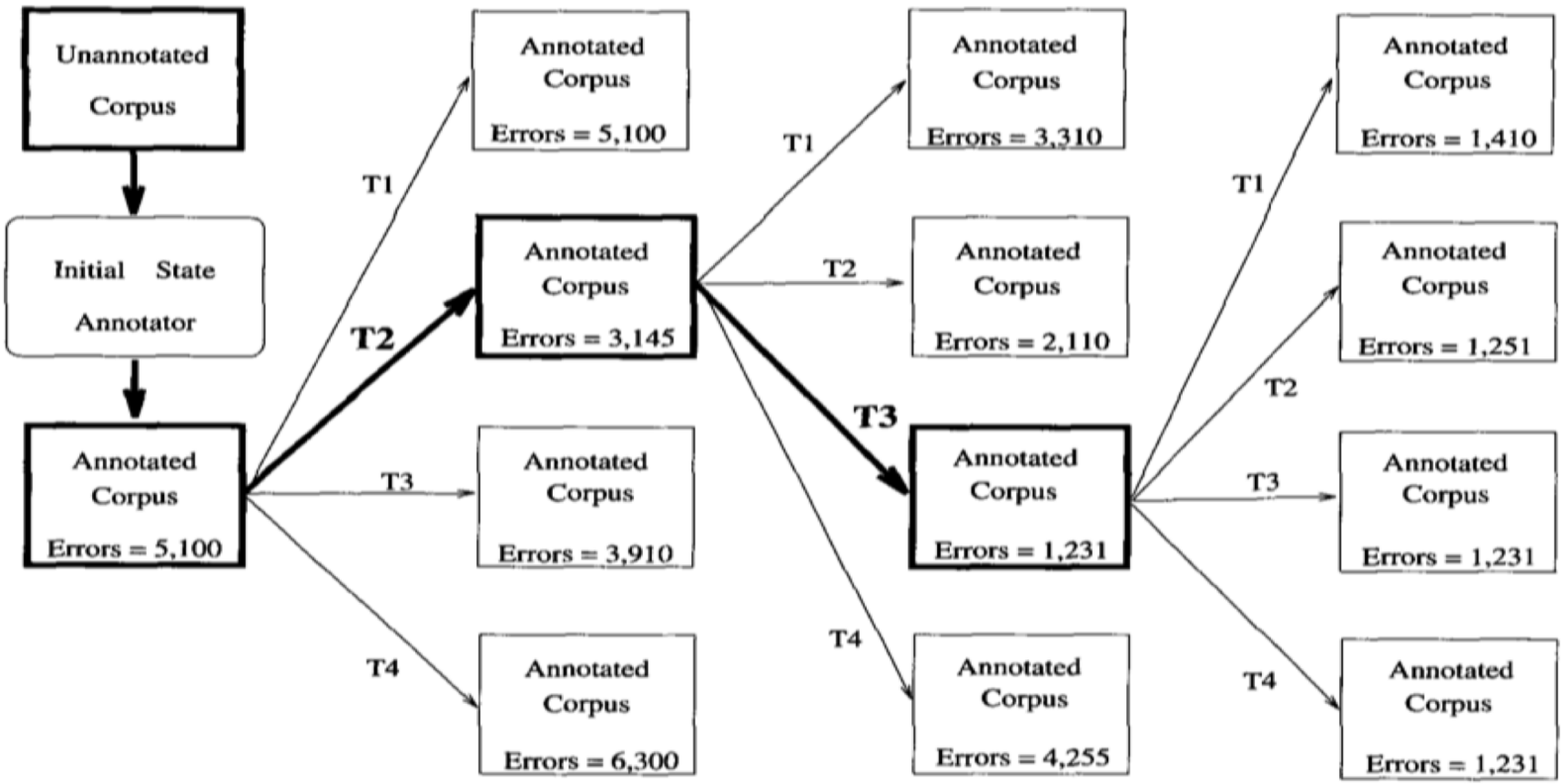


Figure 2
An Example of Transformation-Based Error-Driven Learning.

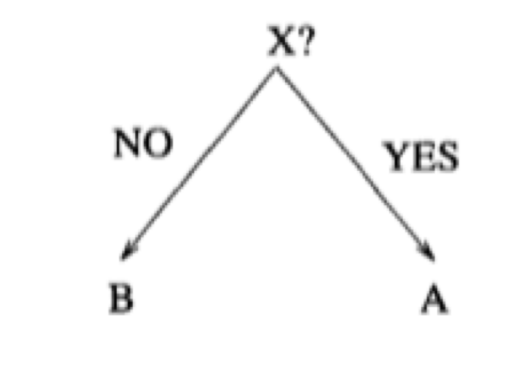
Pour définir des applications spécifiques de l’apprentissage transformé, il faut spécifier:

1. L'annotateur d'état initial
2. L'espace des transformations autorisées (règles de réécriture et environnements déclencheurs)
3. La fonction objective pour comparer le corpus à la vérité et choisir une transformation.

Deux paramètres supplémentaires doivent être spécifiés: l'ordre dans lequel les transformations sont appliquées à un corpus et si une transformation est appliquée immédiatement ou uniquement après que l'ensemble du corpus a été examiné pour les environnements de déclenchement.

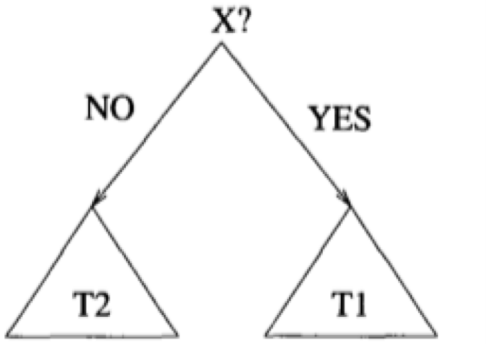
Les arbres de décision = les listes de transformation

Decision Tree



- Etiquette S
- Si X donc S → A
- S → B

Transformation list



T1 → L1 T2 → L2

L1 → S' L2 → S''

- Etiquette S
- Si X donc S → S'
- S → S''

Les arbres de décision ≠ les listes de transformation

Mais les arbres de décision sont utile dans les cas plutôt binaires.

La puissance supplémentaire des listes de transformation provient du fait que les résultats intermédiaires de la classification d'un objet sont reflétés dans l'étiquette actuelle de cet objet, rendant ainsi cette information intermédiaire disponible pour une utilisation dans la classification d'autres objets. Ce n'est pas le cas pour les arbres de décision, où le résultat des questions posées est enregistré implicitement par l'emplacement actuel dans l'arbre.

L’étiquetage morphosyntaxique

Au début l’étiquetage étaient seulement manuelle

Utilité dans les domaines :

- Reconnaissance de la parole
- Traduction automatique
- Analyse syntaxique
- Lexicographie

L'étiquetages morphosyntaxique avec Transformation-Based Error-Driven Learning

- Chaque mot reçoit l'étiquette le plus probable de corpus entrainé par l'étiqueteur initial
- Chaque mot obtient tous les étiquettes possibles mais parmi eux l'un est marqué comme le plus probable:

  Ex: *half* : CD DT JJ NN PDT RB VB

- Ensuite des règles d'après la position des mots dans le texte sont élaborées
- Utilisation des règles et la vérification pour réduire le nombre de fautes

EX : Changez l'étiquette *X* à *Y* si elle est précédé par *Z*

Lexicalisation des étiquettes

- Références non seulement aux tags mais aussi aux mots

Les règles: Remplacez l'étiquette *a* par *b* quand

1. Le mot précèdent est *w*
2. Le deuxième mot avant (après) est le mot *w*
3. Un de deux mots qui précèdent (suivent) le mot est *x*
4. Le mot actuel est *w* et le mot précédé (suivant) est *x*
5. Le mot actuel est *w* et le mot précédé (suivant) est étiqueté *z*
6. Le mot actuel est *w*
7. Le mot actuel est *w* et le mot précédé (suivant) est étiqueté *t*.
8. Le mot actuel est *w*, le mot précédé (suivant) est w_2 est le mot précédé (suivant) est étiqueté *t*

W et x sont les variables parmi tous les mots de corpus et *z* et *t* sont des variables de tous les parties du discours.

Exemple : collocation as … as

Changez le tag de **IN** (préposition) à **RB** (adverbe) si as et le deuxième mot à droite.

As/IN tall/JJ as/IN → As/RB tall/JJ as/IN

L’étiquetages des mots inconnus

Au début : nom propre avec une majuscule et nom commun dans les autres cas

Les règles appliquées :

Remplacez l’étiquette d’un mot inconnu (X) à Y si:

1. Enlèvement du préfixe (suffixe) x, $|x| \leq 4$ donne un mot.
2. Le premier (dernier) (1,2,3,4) caractères de mot est x.
3. Ajout de chaine de caractère x comme préfix (suffixe) donne un mot.
4. Le mot w apparaît immédiatement à gauche du mot.
5. Le caractère z apparaît dans le mot.

#	Change Tag From	To	Condition
1	NN	NNS	Has suffix **-s**
2	NN	CD	Has character **.**
3	NN	JJ	Has character **-**
4	NN	VBN	Has suffix **-ed**
5	NN	VBG	Has suffix **-ing**
6	??	RB	Has suffix **-ly**
7	??	JJ	Adding suffix **-ly** results in a word.
8	NN	CD	The word **$** can appear to the left.
9	NN	JJ	Has suffix **-al**
10	NN	VB	The word **would** can appear to the left.
11	NN	CD	Has character **0**
12	NN	JJ	The word **be** can appear to the left.
13	NNS	JJ	Has suffix **-us**
14	NNS	VBZ	The word **it** can appear to the left.
15	NN	JJ	Has suffix **-ble**
16	NN	JJ	Has suffix **-ic**
17	NN	CD	Has character **1**
18	NNS	NN	Has suffix **-ss**
19	??	JJ	Deleting the prefix **un-** results in a word
20	NN	JJ	Has suffix **-ive**

Figure 6
The first 20 transformations for unknown words.

Example : actress

Remplacez une étiquette du nom commun au pluriel par un nom commun singulier du mot a suffixe -ss.

Taux de précision

Taux de précision a été attesté sur les différents types de corpus et a était très élevé

Corpus	Accuracy
Penn WSJ	96.6%
Penn Brown	96.3%
Orig Brown	96.5%

K-Best tag

# of Rules	Accuracy	Avg. # of tags per word
0	96.5	1.00
50	96.9	1.02
100	97.4	1.04
150	97.9	1.10
200	98.4	1.19
250	99.1	1.50

Les k-best tags sont attribués dans un tagueur stochastique en renvoyant toutes les tags dans les limites d'un seuil de probabilité d'être correct pour un mot particulier. Au lieu de changer l'étiquetage d'un mot, les transformations ajoutent maintenant des étiquettes alternatives à un mot.

Example : changez l'étiquette X en étiquette Y → ajouter l'étiquette X à l'étiquette Y

ajoutez l'étiquette X au mot W (ajout d’une étiquette alternative)

Merci !