Übersicht Kapitel 10

Part-of-Speech Tagging

- Informationsquellen beim Tagging
- Markov Model Taggers
- Hidden Markov Model Taggers
- Transformationsbasiertes Tagging

Tagging

Tagging ist die Tätigkeit jedes Wort in einem Satz mit der entsprechenden Wortart zu kennzeichnen.

Tag	Part Of Speech
AT	article
BEZ	the word *is*
IN	preposition
JJ	adjective
JJR	comparative adjective
MD	modal
NN	singular or mass noun
NNP	singular proper noun
NNS	plural noun
PERIOD	. : ? !
PN	personal pronoun
RB	adverb
RBR	comparative adverb
TO	the word *to*
VB	verb, base form
VBD	verb, past tense
VBG	verb, present participle, gerund
VBN	verb, past participle
VBP	verb, non-3rd person singular present
VBZ	verb, 3rd singular present
WDT	*wh-* determiner (*what*, *which*)

Table 10.1 Some part-of-speech tags frequently used for tagging English.

Informationsquellen beim Tagging

Es gibt zwei Informationsquellen:

- SYNTAGMATIC STRUCTURAL INFORMATION
  - Informationen über Tagfolgen werden berücksichtigt.
  - An sich nicht sehr erfolgreich (Bsp. Nur 77% korrekt gesetzte Tags, Greene and Rubin)
- LEXICAL INFORMATION
  - Vorhersagen eines Tags anhand von Informationen über das betreffende Wort.
  - 'Dumb'-tagger (von Charniak, bestimmt das am meisten übliche Tag für jedes Wort) bestimmte 90% aller Tags korrekt.

Visible Markov Model Taggers

Die 'States' eines Markov Modells sind hier die Tags.

w_i	the word at position i in the corpus
t_i	the tag of w_i
$w_{i,i+m}$	the words occurring at positions i through $i+m$
	(alternative notations: $w_i \cdots w_{i+m}$, $w_i, \ldots, w_{i+m}$, $w_{i(i+m)}$)
$t_{i,i+m}$	the tags $t_i \cdots t_{i+m}$ for $w_i \cdots w_{i+m}$
w^l	the l^{th} word in the lexicon
t^j	the j^{th} tag in the tag set
$C(w^l)$	the number of occurrences of w^l in the training set
$C(t^j)$	the number of occurrences of t^j in the training set
$C(t^j, t^k)$	the number of occurrences of t^j followed by t^k
$C(w^l : t^j)$	the number of occurrences of w^l that are tagged as t^j
T	number of tags in tag set
W	number of words in the lexicon
n	sentence length

Table 10.2 Notational conventions for tagging.

Visible Markov Model Taggers

- 'Maximum likelihood' Abschätzung für Tag t^k folgt auf tag t^j:

$$P(t^k|t^j) = \frac{C(t^j, t^k)}{C(t^j)}$$

- 'Maximum likelihood' Abschätzung für W-keit dass ein Wort von einem Tag ausgegeben wird:

$$P(w^l|t^j) = \frac{C(w^l, t^j)}{C(t^j)}$$

- Ermitteln des besten Taggings $t_{1,n}$ für einen Satz $w_{1,n}$:

$$\arg_{t_{1,n}} maxP(t_{1,n}|w_{1,n}) = arg_{t_{1,n}} maxP(w_{1,n}|t_{1,n})P(t_{1,n})$$

Visible Markov Model Taggers

Zusätzlich zu Limited Horizon, werden zwei weitere Voraussetzungen für Wörter gemacht:

- Wörter sind voneinander unabhängig
- Die Identität eines Worts hängt ausschliesslich von seinem Tag ab

$$P(w_{1,n}|t_{1,n})P(t_{1,n}) = \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Finale Gleichung für das Ermitteln der optimalen Tags für einen Satz:

$$\hat{t}_{1,n} = arg_{t_{1,n}} max P(t_{1,n}|w_{1,n}) = arg_{t_{1,n}} max \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Ein effizienter Tagging-Algorithmus ist der Viterbi Algorithmus (Kapitel 9).

Viterbi Algorithmus

comment: Given: a sentence of length n
comment: Initialization
$\delta_1(\text{PERIOD}) = 1.0$
$\delta_1(t) = 0.0 \quad \text{for } t \neq \text{PERIOD}$
comment: Induction
for $i := 1$ **to** n **step** 1 **do**
    for all tags t^j **do**
        $\delta_{i+1}(t^j) := \max_{1 \leq k \leq T} [\delta_i(t^k) \times P(t^j|t^k) \times P(w_{i+1}|t^j)]$
        $\psi_{i+1}(t^j) := \arg\max_{1 \leq k \leq T} [\delta_i(t^k) \times P(t^j|t^k) \times P(w_{i+1}|t^j)]$
    end
end
comment: Termination and path-readout
$X_{n+1} = \arg\max_{1 \leq j \leq T} \delta_{n+1}(j)$
for $j := n$ **to** 1 **step** -1 **do**
    $X_j = \psi_{j+1}(X_{j+1})$
end
$P(X_1, \ldots, X_n) = \max_{1 \leq j \leq T} \delta_{n+1}(t^j)$

Figure 10.2 Algorithm for tagging with a Visible Markov Model Tagger.

Variationen

- Unknown words
- Trigram taggers
- Interpolation and variable memory
- Smoothing
- Reversibility
- Sequence vs. tag by tag

Hidden Markov Model Taggers

Existieren keine Trainingsdaten, kann man ein HMM nutzen um Regelmäßigkeiten von Tagfolgen zu lernen.

Üblicherweise wird 'dictionary information' benutzt um die Parameter eines Modells einzuschränken.

Dafür gibt es zwei Methoden:

- Methode von Jelinek
  - 'Word generation probabilities' für ein Wort auf null setzen wenn das dazugehörige Word-Tag-Paar nicht im Dictionary aufgeführt ist. (z.B. JJ nicht möglich für 'book')
- Methode von Kupiec
  - Wie die Methode von Jelinek, jedoch werden hier erst alle Wörter in äquivalente Wort-Klassen gruppiert, sodass alle Wörter für die das selbe Tagset möglich ist in der selben Wort-Klasse sind.

Transformationsbasiertes Tagging

Ein anfänglich (teilweise) unkorrektes Tagging wird in ein Tagging mit weniger Fehlern transformiert.

Zwei Schlüsselkomponenten:

- Eine Spezifikation die vorgibt welche 'Fehler-Korrektur'-Transformationen zulässig sind und
- der dazugehörige Lernalgorithmus

Input: Ein 'getaggter' Korpus und ein Dictionary.

Transformationsbasiertes Tagging

Eine Transformation besteht aus zwei Teilen:

- einer Auslöserumgebung ('triggering environment'), siehe Tabelle 10.7 und
- einer 'rewrite rule' ($t^1 \rightarrow t^2$ 'ersetze tag t^1 mit tag t^2')

Beispiele in Tabelle 10.8

Der Lernalgorithmus bestimmt die besten Transformationen und bestimmt deren Ausführungsreihenfolge (Bild 10.3).