Word Sense Disambiguation; WSD

@ampeloss

December 13, 2013

語は曖昧性を持つ. . . ex. ”bank” 提喩 etc..

曖昧性解消 (Word Sense Disambiguation ; WSD) とは
文脈を使った語義の判断のこと

- 例えば翻訳で必要なタスクである

index

準備

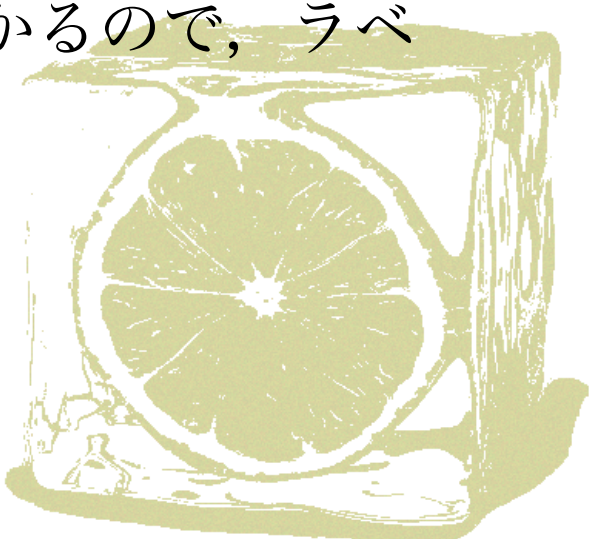
- 教師アリと教師ナシ
- 擬似語
- 正解率の上限・下限

教師アリと教師ナシ

- 教師アリ学習はクラスタリング (Chap.16) あるいは
  フィッティング問題と見做せる

教師アリと教師ナシ

- 教師アリ学習はクラスタリング (Chap.16) あるいは
  フィッティング問題と見做せる
- NLP に於いてラベル付けはコストがかかるので，ラベ
  ルのついてないデータから
  学習したい (これは豊富にある)



pseudo word

テスト，評価の為には曖昧語を含む文章を大量に用意したいけど大変なので擬似語を用いる

擬似語とは人工的に作った，曖昧な意味を与えた言葉

ex. banana-word

実際には
擬似語とオリジナルの両方の
曖昧性解消を評価する

上限と下限

アルゴリズムの良さを示すのに正解率を示す
ここで上限と下限を知らないと良いのかよくわからない

上限

- 上限は普通人手でつけた正解率を使う
- Gale らが示した上限は 97-99 %

下限

- Baseline と呼ばれるもの
- 可能な限りアルゴリズムを
  単純化した時の正解率を使う
- 90 % くらいが普通

変数

以下のような意味で変数を用いる

- w　　曖昧語
- s_k　　曖昧語の意味 (sense)
- c　　曖昧語の文脈 (context)
  普通は，曖昧語の周辺何 word とか
- v_i　　文脈の中の語 (vocabulary)

教師アリ

未知語 w に対してラベル s がついてる

- 単純ベイズ分類器 (by Gale ら)
- 情報理論アプローチ (by Brown ら)

単純ベイズ分類器

語義の選択にベイズ決定理論を用いる

ベイズ決定理論

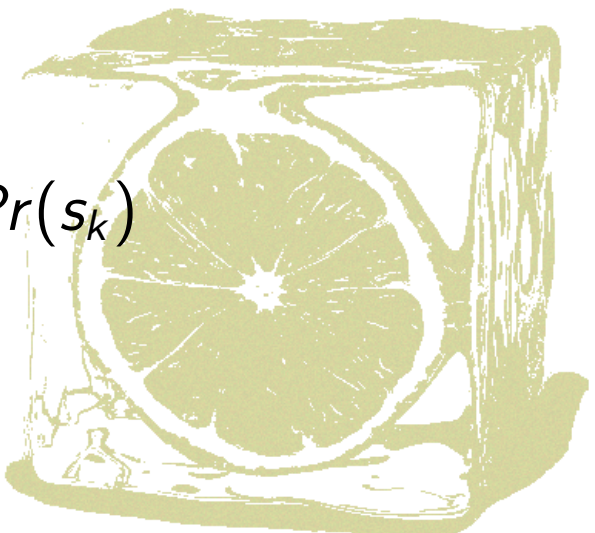
$$s = \underset{s_k}{\operatorname{argmax}}\, Pr(s_k|c)$$

where

- s_k is a sense
- c is a context

ベイズ決定理論

$$
\begin{aligned}
s &= \underset{s_k}{\operatorname{argmax}}\, Pr(s_k|c) \\
&= \underset{s_k}{\operatorname{argmax}}\, \frac{Pr(c|s_k)}{Pr(c)} Pr(s_k) \\
&= \underset{s_k}{\operatorname{argmax}}\, \log Pr(c|s_k) + \log Pr(s_k)
\end{aligned}
$$



単純ベイズ仮定

$$
\begin{aligned}
s &= \underset{s_k}{\operatorname{argmax}} \log Pr(c|s_k) + \log Pr(s_k) \\
&= \underset{s_k}{\operatorname{argmax}} \sum_{v \in c} \log Pr(v|s_k) + \log Pr(s_k)
\end{aligned}
$$

単純ベイズ仮定

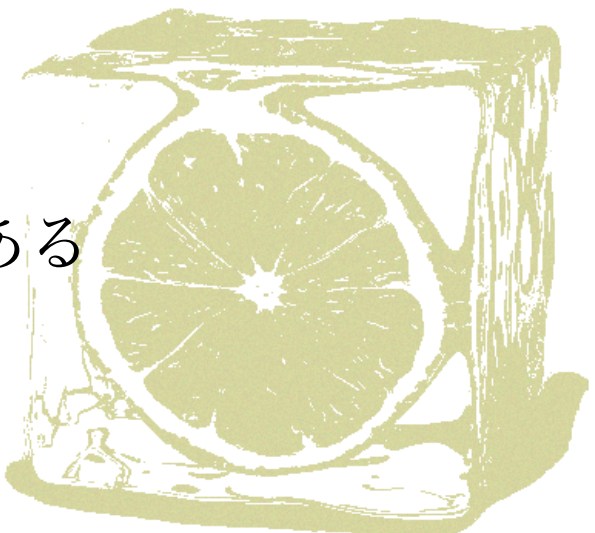
$$
\begin{aligned}
s &= \underset{s_k}{\operatorname{argmax}} \log Pr(c|s_k) + \log Pr(s_k) \\
&= \underset{s_k}{\operatorname{argmax}} \sum_{v \in c} \log Pr(v|s_k) + \log Pr(s_k)
\end{aligned}
$$

- c における v の独立性を仮定してる

単純ベイズ仮定

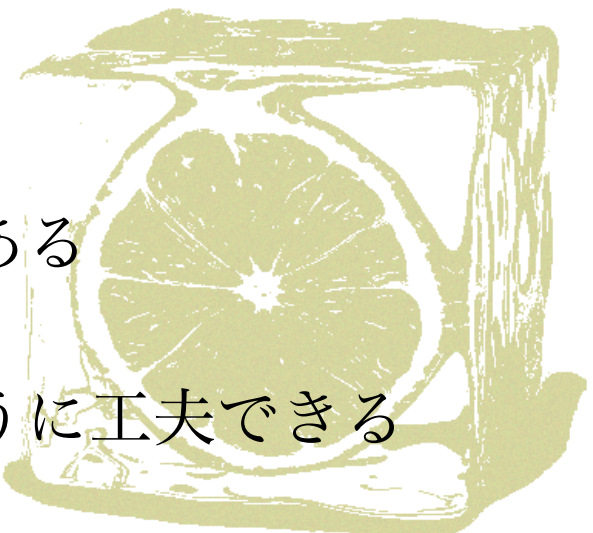
$$
\begin{aligned}
s &= \operatorname*{argmax}_{s_k} \log Pr(c|s_k) + \log Pr(s_k) \\
&= \operatorname*{argmax}_{s_k} \sum_{v \in c} \log Pr(v|s_k) + \log Pr(s_k)
\end{aligned}
$$

- c における v の独立性を仮定してる
- 確かにこの仮定が成り立たないこともある
  ex. c = { election , president }



単純ベイズ仮定

$$
\begin{aligned}
s &= \operatorname*{argmax}_{s_k} \log Pr(c|s_k) + \log Pr(s_k) \\
&= \operatorname*{argmax}_{s_k} \sum_{v \in c} \log Pr(v|s_k) + \log Pr(s_k)
\end{aligned}
$$



- c における v の独立性を仮定してる
- 確かにこの仮定が成り立たないこともある
  ex. c = { election , president }
- 実際にはほとんどの場合うまくいくように工夫できる
  (Domingos and Pazzani 1997)

$Pr(v|s)$ とか $Pr(s)$ は MLE で求めるならば
訓練データ中で数えるだけ
すなわち

$$Pr(v|s) = \frac{C(v,s)}{C(s)}$$

$$Pr(s) = \frac{C(s)}{C(w)}$$

情報理論によるアプローチ

単純ベイズ仮定が効かなそうな時の為のアプローチ
(Brown et al. 1991)

曖昧語について，indicator を用いて意味を判別できる
indicator とは

- 目的語
- 文の時制
- すぐ左に置かれてる語が何であるか

などという値

情報理論によるアプローチ

例えば，仏語 prendre という動詞は
目的語が

- measure だったら take
- decision だったら make

という語義を持つ

情報理論によるアプローチ

一つの曖昧語について
翻訳 $\{t_1, \ldots, t_n\}$
indicator の値 $\{x_1, \ldots, x_m\}$

例えば，prendre という曖昧語について
t1 = take , t2 = make
x1 = measure , x2 = decision

Flip-Flop アルゴリズム

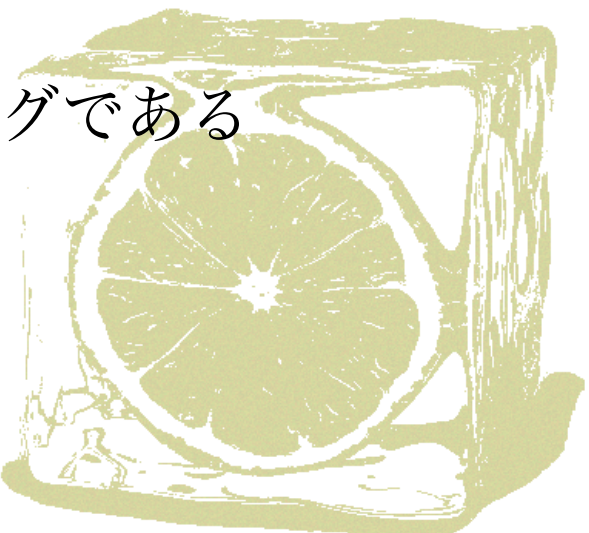
このアルゴリズムは翻訳 $\{t_i\}$ を 2 つに分割 (それぞれをクラスと呼ぶ)
indicator をそれぞれどちらかのクラスに対応付けるように 2 つに分割する

$\max I(P; Q)$
  where
    $P = \{P1, P2\} = partitionOf\{t_i\}$
    $Q = \{Q1, Q2\} = partitionOf\{x_j\}$

Flip-Flop アルゴリズム

loop (P = random) =
  Q ← arg-max[Q] I(P;Q)
  P' ← arg-max[P] I(P;Q)
  if P == P'
    then return (P', Q)
    else loop(P')

以上が教師アリ学習
すなわち，曖昧語について sense というラベルを付与したデータを必要とする．しかしながら最後の Brown らによるアルゴリズムの中の indicator はラベルであると考えることもできる
そうすると，ラベルから sense へのマッピングである



Dictionary-Based Disambiguation

辞書またはシソーラスを用いる手法
使い方にも色々あって

- sense definitions
- semantic categorization of words
- bilingual dictionary

語義定義に基づく WSD

辞書ベースによる WSD の中で一番簡単なもの (Lesk，1986)

例えば"ash" という語に対して次のような 2 つの定義がある

- sense1
  - tree
  - a tree of the olive family
- sense2
  - burned stuff
  - the solid residue left when combustible material is burned

例文: This cigar burns slowly and creates a stiff ash .

例文: This cigar burns slowly and creates a stiff ash .

- the solid residue left when combustible material is burned (sense 2)

例文: This cigar burns slowly and creates a stiff ash .

- a tree of the olive family (sense 1)
- the solid residue left when combustible material is burned (sense 2)

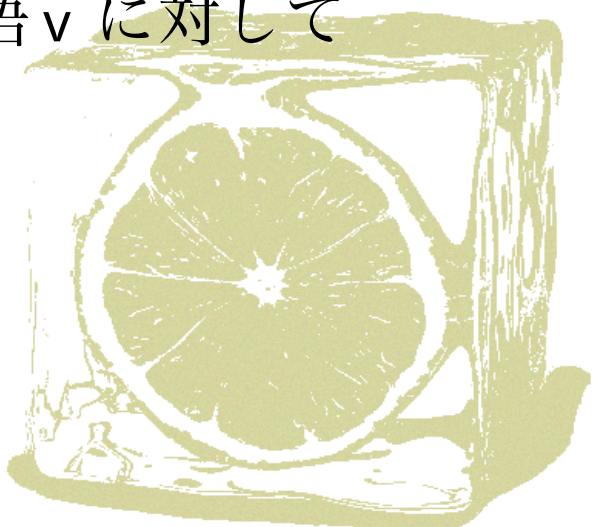
例文: This cigar burns slowly and creates a stiff ash .

- a tree of the olive family (sense 1)
- the solid residue left when combustible material is burned (sense 2)
- ⇒ sense 2 を選択

Leak's algorithm

曖昧語 w に対して
意義 $\{s_1, \ldots, s_K\}$ があったとして
定義の文 $\{D_1, \ldots, D_K\}$ とする
w のコンテキスト c に現れるそれぞれの単語 v に対して

$$E_v = \bigcup \{D_k | v \in D_k\}$$



let c = context of w

let k = $\underset{1 \leq k \leq K}{\operatorname{argmax}} |D_k \cap \{E_v | v \in c\}|$

s_k

where

D_k = bag of Definition for s_k

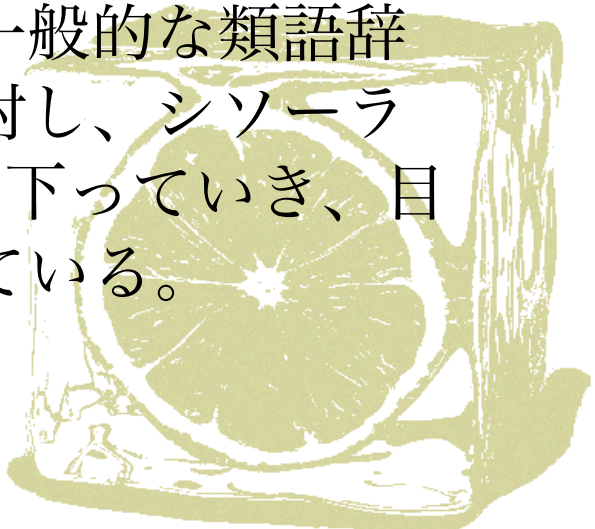
$E_v = \bigcup \{D_k | v \in D_k\}$

let c = context of w
let k = $\underset{1 \leq k \leq K}{\operatorname{argmax}} |D_k \cap \{E_v | v \in c\}|$
s_k
where
D_k = bag of Definition for s_k
$E_v = \bigcup \{D_k | v \in D_k\}$

- Lesk さんによると正解率は 50-70 %

シソーラスを用いたWSD

シソーラス (Thesaurus) とは、単語の上位/ 下位関係、部分/ 全体関係、同義関係、類義関係などによって単語を分類し、体系づけた辞書。
[..]
シソーラスは類語辞典の一種といえるが、一般的な類語辞典は五十音順に項目立てがされているのに対し、シソーラスは語彙の持つ意味から、大分類- 中分類と下っていき、目的の単語に達することができるようになっている。
(by ja.wikipedia)



Walker 's thesaurus-based algorithm

単純なシソーラスの使用例

Walker 's thesaurus-based algorithm

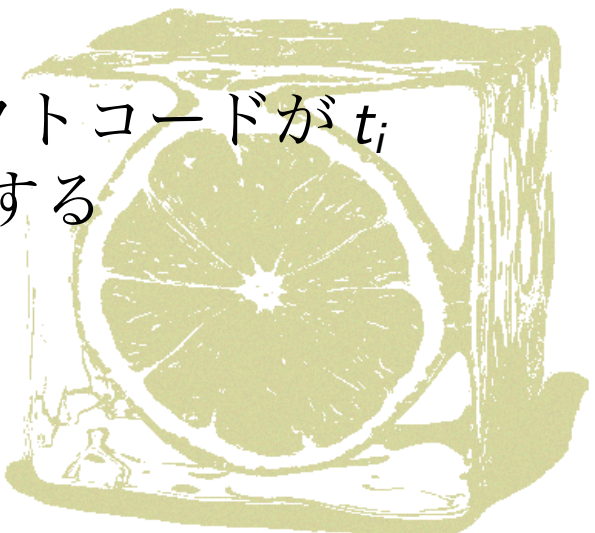
単純なシソーラスの使用例

- 曖昧語 w の現れるコンテキスト c
- w の語義 s_i に対してシソーラスによるサブジェクトコード t_i とする

Walker 's thesaurus-based algorithm

単純なシソーラスの使用例

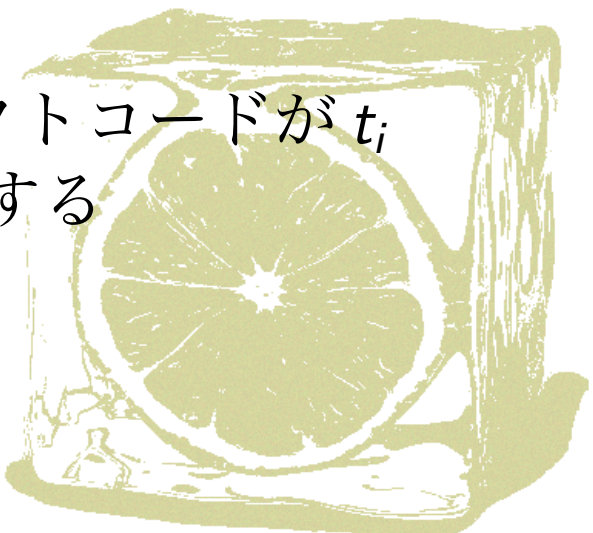
- 曖昧語 w の現れるコンテキスト c
- w の語義 s_i に対してシソーラスによるサブジェクトコード t_i とする
- 文脈 c の中に現れる語の内，サブジェクトコードが t_i であるものの個数を語義 s_i のスコアとする



Walker 's thesaurus-based algorithm

単純なシソーラスの使用例

- 曖昧語 w の現れるコンテキスト c
- w の語義 s_i に対してシソーラスによるサブジェクトコード t_i とする
- 文脈 c の中に現れる語の内，サブジェクトコードが t_i であるものの個数を語義 s_i のスコアとする
- スコアが最大となる語義を選択する



Black (1998) さんは interest, point, power, state, terms という単語の曖昧性解消を 50 % の正解率で実現できた
本曰く，その 5 つの単語は大変曖昧性解消が難しいので 50 % というのはすごい

Walker 's algorithm の問題点

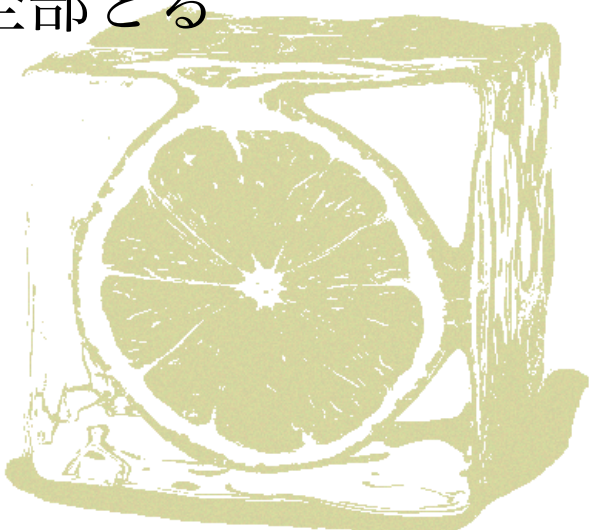
コンピュータのマニュアルにおいて「マウス」という言葉はシソーラスにおける哺乳類」を表すための手がかりにならない

Yarowsky 's algorithm

- 曖昧語 w の語義 s_i に対する
  シソーラスカテゴリを t_i とする

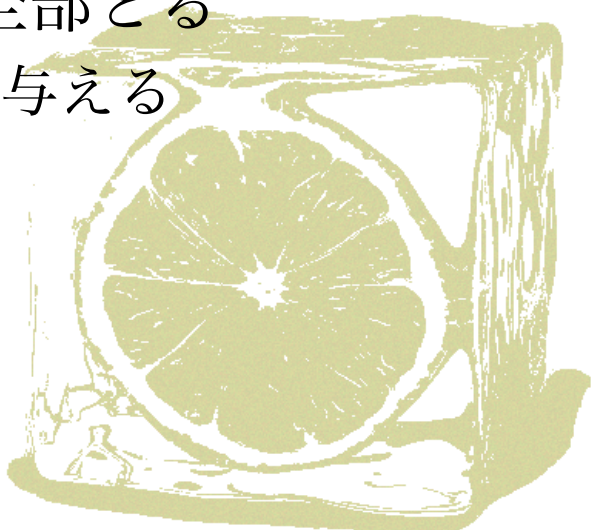
Yarowsky 's algorithm

- 曖昧語 w の語義 s_i に対する
  シソーラスカテゴリを t_i とする
- 語を中心に 100 語を一つのコンテキスト c_i とする
  コーパス全体に対してコンテキストを全部とる



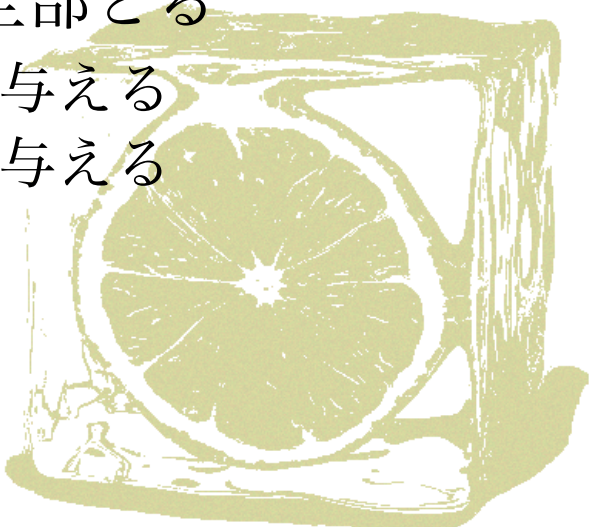
Yarowsky 's algorithm

- 曖昧語 w の語義 s_i に対する
  シソーラスカテゴリを t_i とする
- 語を中心に 100 語を一つのコンテキスト c_i とする
  コーパス全体に対してコンテキストを全部とる
  - コンテキストに語によってカテゴリを与える



Yarowsky 's algorithm

- 曖昧語 w の語義 s_i に対する
  シソーラスカテゴリを t_i とする
- 語を中心に 100 語を一つのコンテキスト c_i とする
  コーパス全体に対してコンテキストを全部とる
  - コンテキストに語によってカテゴリを与える
  - 語にコンテキストによってカテゴリを与える



Yarowsky 's algorithm

- 曖昧語 w の語義 s_i に対する
  シソーラスカテゴリを t_i とする
- 語を中心に 100 語を一つのコンテキスト c_i とする
  コーパス全体に対してコンテキストを全部とる
  - コンテキストに語によってカテゴリを与える
  - 語にコンテキストによってカテゴリを与える
  - 曖昧語を含むコンテキストに語によってカテゴリを与える

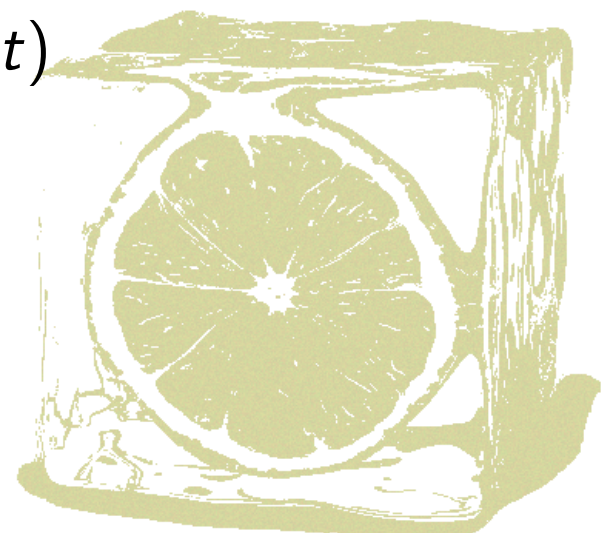
コンテキスト c に対して
ありうるシソーラスカテゴリ t を集める

- $$Pr(t|c) = \frac{Pr(c|t)}{Pr(c)} Pr(t)$$

コンテキスト c に対して
ありうるシソーラスカテゴリ t を集める

- $$Pr(t|c) = \frac{Pr(c|t)}{Pr(c)} Pr(t)$$
- 単純ベイズ仮定

$$Pr(t|c) = \prod_{v \in c} \frac{Pr(v|t)}{Pr(v)} Pr(t)$$

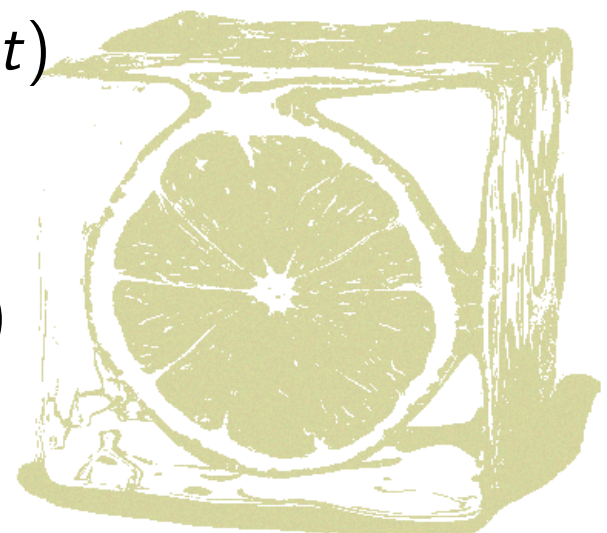


コンテキスト c に対して
ありうるシソーラスカテゴリ t を集める

- $$Pr(t|c) = \frac{Pr(c|t)}{Pr(c)} Pr(t)$$
- 単純ベイズ仮定

$$Pr(t|c) = \prod_{v \in c} \frac{Pr(v|t)}{Pr(v)} Pr(t)$$

- $$score(c, t) = \log Pr(t|c)$$



コンテキスト c に対して
ありうるシソーラスカテゴリ t を集める

- $$Pr(t|c) = \frac{Pr(c|t)}{Pr(c)} Pr(t)$$
- 単純ベイズ仮定

$$Pr(t|c) = \prod_{v \in c} \frac{Pr(v|t)}{Pr(v)} Pr(t)$$

- $$score(c, t) = \log Pr(t|c)$$
- スコアが一定の水準 α を超えたカテゴリの集合を
  そのコンテキスト c がありうるカテゴリ $t(c)$ とする

- カテゴリ t_j に対してトピックとは

$$T_j = \{c_i | t_i \in t(c_i)\}$$

- 語 v_j に対して語彙とは

$$V_j = \{c_i | v_j \in c_i\}$$

次のそれぞれの確率の以下の通りに定める (もう意味不明)

$$Pr(v_j|t_k) = \frac{|V_j \cap T_i|}{(\sum_j |V_j \cap T_i|)}$$

$$Pr(t_i) = \frac{(\sum_j |V_j \cap T_i|)}{(\sum_i \sum_j |V_j \cap T_i|)}$$

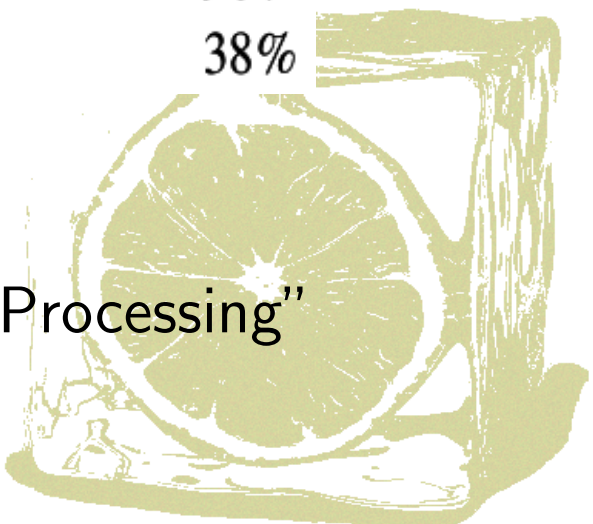
先の 2 つの確率によって，w の語義が s_k である尤度

$$score(s_k) = \log\left(Pr(t|c)Pr(c)\right)$$

これが最大になる語義 s_k を選択すればよい

Word	Sense	Roget category	Accuracy
bass	musical senses	MUSIC	99%
	fish	ANIMAL, INSECT	100%
star	space object	UNIVERSE	96%
	celebrity	ENTERTAINER	95%
	star shaped object	INSIGNIA	82%
interest	curiosity	REASONING	88%
	advantage	INJUSTICE	34%
	financial	DEBT	90%
	share	PROPERTY	38%

Table 7.6
"Foundations of Statistical Natural Language Processing"
247 ページより引用



第二言語への翻訳に基づく WSD

元の言語を第一言語として第一言語から第二言語への翻訳と，
第二言語のコーパスを利用する
(Dagan et al. 1991 ; Dagan and Itali 1994)

英単語 interest は 2 つの独単語 Beteilingung, Interesse への訳がある showed interest という使われ方があった時

- show → zeigen
- zeigen Interesse とは言う
- zeigen Beteilingung とは言わない

その interest は Interesse の語義だと分かる

先の例だと曖昧語が目的語でそれに対する動詞という関係をR として

$$
\begin{aligned}
score(s) &= |\{c|R(w,v), w' \in T(w), v' \in T(v), R(w',v')\}| \\
s' &= \mathrm{argmax}\, score(s)
\end{aligned}
$$

Dagan and Itali の実験はもう少し複雑なことをしていてヘブライ語の ro'sh (head, top) を 90 % 以上で正しく判別できた

語義の一貫性

- One sense per discouse
- One sense per collocation

- 曖昧語 w が複数回登場するドキュメントについて

- 曖昧語 w が複数回登場するドキュメントについて
- 小さい単位に区切って (区切り方の議論は section 15.5)
  それぞれの w について語義を選択

- 曖昧語 w が複数回登場するドキュメントについて
- 小さい単位に区切って (区切り方の議論は section 15.5)
  それぞれの w について語義を選択
- 多数決をとって全ての w の語義をそれに改める

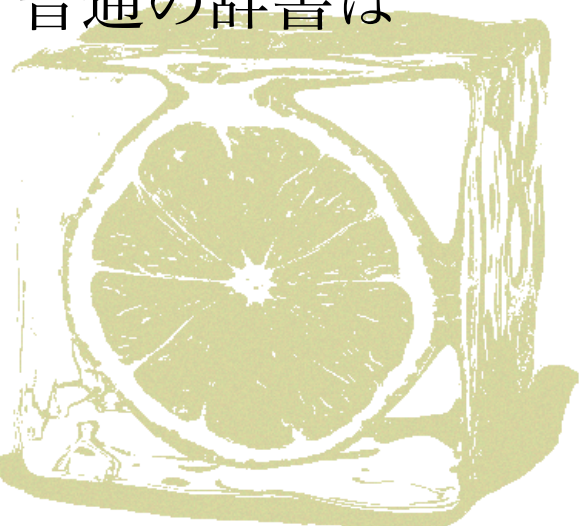
教師ナシ学習

どのような場合に必要か

教師ナシ学習

どのような場合に必要か

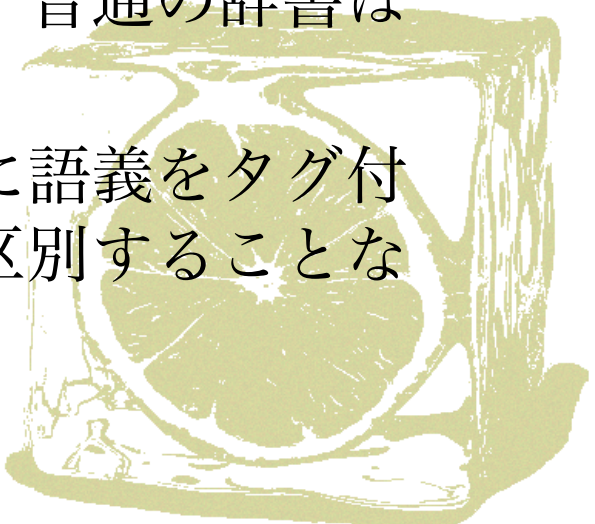
- 訓練データが得られない場合
- 辞書が得られない場合
- 例えば特殊なドメインを対象とする時，普通の辞書は無意味



教師ナシ学習

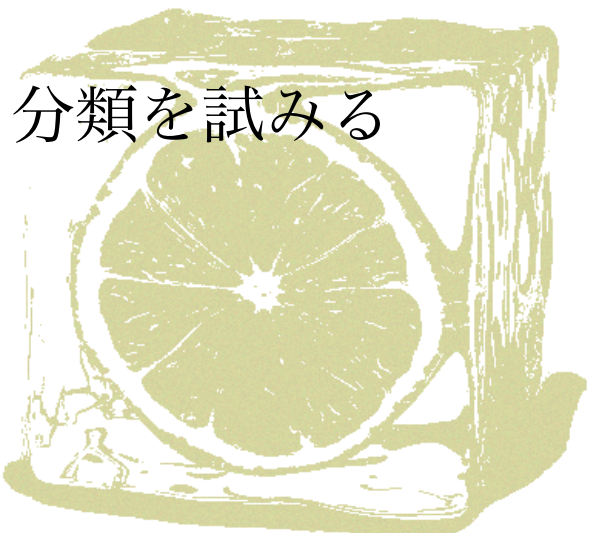
どのような場合に必要か

- 訓練データが得られない場合
- 辞書が得られない場合
- 例えば特殊なドメインを対象とする時，普通の辞書は無意味
- 訓練データが無い以上，もちろん本当に語義をタグ付けするようなことは不可能であるが，区別することならできる



曖昧語 w
w の語義 $s_1 \ldots s_K$
w の文脈 c_i
文脈 c_i 中の語 v_j

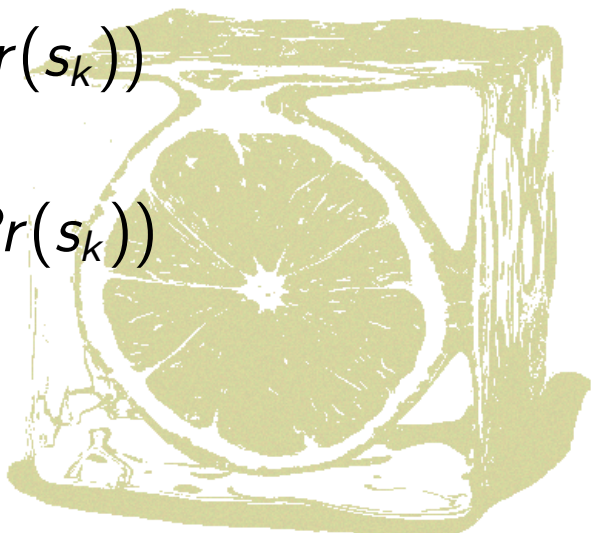
語義の数 K を例えば 20 とか多めに取って，分類を試みる



EM algorithm

コーパス C
モデル $\mu = (Pr(v_j|s_k), Pr(s_k))$
　に対して次の対数尤度

$$
\begin{aligned}
I(C|\mu) &= \log\left(\prod_i \sum_k Pr(c_i|s_k)Pr(s_k)\right) \\
&= \sum_i \log\left(\sum_k Pr(c_i|s_k)Pr(s_k)\right)
\end{aligned}
$$



これを最大化するモデル μ を構成したい

$Pr(v_j|s_k)$ 及び $Pr(s_k)$ にランダムな数値を割り当てることから始め

l が収束するまで E-step M-step を繰り返す

loop:
    E-step; M-step; loop

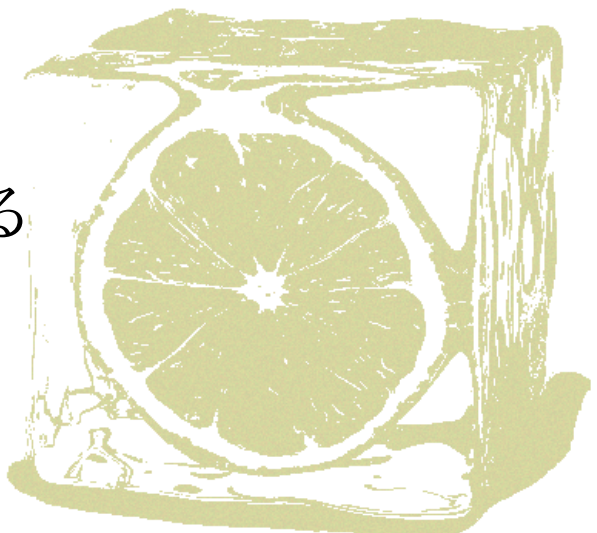
E-step

– 期待値ステップ

for all i, k

$$h_{ik} \leftarrow \frac{Pr(c_i|s_k)}{\Sigma_k Pr(c_i|s_k)}$$

$Pr(c_i|s_k)$ の算出には単純ベイズ仮定を用いる



M-step

– 最大化ステップ

$$Pr(v_j|s_k) \leftarrow \frac{1}{Z} \sum_{v_j \in c_i} h_{ik}$$

$$Pr(s_k) \leftarrow \frac{1}{J} \sum_{i} h_{ik}$$

Z と J は確率の総和が 1 に成るための適当な数

構成できたモデル μ を用いて

$$s' = \underset{s_k}{\operatorname{argmax}}\, Pr(s_k|c)$$

- EM アルゴリズムによる結果は 256 ページ Table7.9
- 辞書ベースでの正解率より 5-10 % 低い程度