自然言語処理 116－4
（1996. 11. 18）

最大エントロピー法を用いた単語bigramの推定

白井清昭, 乾健太郎, 徳永健伸, 田中穂積
東京工業大学大学院情報理工学研究科
{kshirai,inui,take,tanaka}@cs.titech.ac.jp

単語bigramを推定する際には，一般にデータがスパースであるので，何らかのスムージングが必要となる．本稿では，シソーラスを利用した単語の抽象化によるスムージングの手法を分類し，それらとスムージングの結果得られる単語bigramの品質との関係について考察した．そして，最大エントロピー法による確率モデルの推定方法が単語bigramの推定に有効であることに着目した．また，最大エントロピー法をそのまま単語bigramの推定に利用するには計算量が大きいという問題点を解決するために，これを高速化するいくつかの手法を提案した．推定した単語bigramを用いて複合名詞の解析実験を行い，提案した手法の有効性を確認した．

A Maximum Entropy Model for Estimating Lexical Bigrams

SIRAI Kiyoaki, INUI Kentaro,
TOKUNAGA Takenobu, TANAKA Hozumi
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
{kshirai,inui,take,tanaka}@cs.titech.ac.jp

To estimate lexical bigram distribution, smoothing methods are generally applied to overcome concerns of data sparseness. First, we discuss the relation between smoothing methods using a thesaurus and the quality of lexical bigram distribution estimated by them. We conclude that maximum entropy methods are suitable for estimating lexical bigram distribution. As the computational cost associated with maximum entropy methods is too large for lexical bigram distribution estimation, we propose several methods to suppress the overhead. We conducted an experiment to analyze Japanese compound nouns, and the result indicated that our method is better than existing smoothing methods.

1 はじめに

近年，統計情報を利用した自然言語処理に関する研究が盛んに行われている．統計的自然言語処理の特長としては，様々な解析結果の候補に対して順位付けを行うことによる曖昧性解消が行えること，また曖昧性解消に必要な知識をコーパスなどの言語データから自動的に獲得できることなどがあげられる．自然言語処理によく利用される統計的知識の1つとして，単語の共起情報が挙げられる．例えば，文の生成確率の計算には単語または品詞間の隣接 bigram が，統語的曖昧性の解消には動詞とその格要素となる名詞の共起情報が，複合名詞の係り受け関係の解析には名詞間の共起情報が，それぞれ利用できる．本稿においては，このような2単語間の共起の強さを表わす確率モデルを一般に単語 bigram と呼ぶ．

統計的自然言語処理における主な問題点の1つとしてデータスパースネス問題が挙げられる．特に，単語 bigram の場合は推定するパラメタの数が非常に多く，これを言語データから直接推定することは困難である．そこで，何らかのスムージングが必要となる．一般的なスムージングの手法としては，シソーラスを利用して単語を意味クラスに抽象化する方法が挙げられる．本稿では，意味クラスを用いたスムージング方法の違いとその結果得られる単語 bigram の精度について考察する．また，最大エントロピー法による確率モデルの推定が単語 bigram のスムージングに適していることに着目し，単語 bigram を最大エントロピー法を用いて推定する際に生じる問題点を解決する方法を提案する．

2 単語 bigram のスムージング

単語 bigram のスムージングを行う際には，シソーラスなどを利用して単語の意味クラスを導入する方法が一般的である．シソーラスを用いずに単語 bigram のスムージングを行う方法としては，式 (1) のように全ての単語の共起頻度 $C(w_i, w_j)$ に定数γを加える方法や，

$$P(w_i, w_j) = \frac{C(w_i, w_j) + \gamma}{\sum_{i,j}(C(w_i, w_j) + \gamma)} \qquad (1)$$

訓練データにおける確率分布が類似している単語をマージしてスムージングを行う方法などがある [2]．しかしながら，シソーラスなどの他の言語知識を全く使わないよりも，既存の言語知識を積極的に利用した方が，スムージングの精度も向上すると予想される．

シソーラスを利用して単語 bigram のスムージングを行う場合には，式 (2) に示すように，同じ意味クラス C に属する単語 w_i の確率は全て等しいとみなすのが一般的である．

$$P(w_i, w_j) = \frac{1}{|C|} \sum_{w_i \in C} \hat{P}(w_i, w_j) \qquad (2)$$

$\hat{P}(w_i, w_j)$ は訓練データにおける (w_i, w_j) の共起確率である．これは，図 1 の斜線部内の事象の確率 $P(w_i, w_j)$ は全て等しいとし，かつその総和を訓練データと一致させることを意味する．

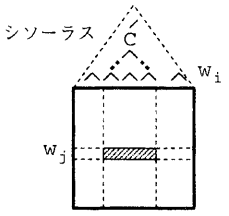

図 1: 意味クラス C によるスムージング

このとき，単語 bigram のスムージングの精度はどのような意味クラスを導入するかに大きく依存する．抽象化レベルの高い (粗い) 意味クラスを用いた場合，推定するパラメタの数は減少するが，多くの単語を1つの意味クラスにまとめてしまうことによる誤差も大きくなる．逆に抽象化レベルの低い (細かい) 意味クラスを用いた場合，抽象化に伴う誤差は小さいが，推定するパラメタの数は多くなる．したがって，シソーラスを用いた単語 bigram のスムージングは，単語の抽象化に伴う誤差をできるだけ小さくし，かつ推定するパラメタの数が少なくなるような意味クラスの集合を求める問題とみなすことができる．

このような最適な意味クラスの集合を決定する問題の難しさや，スムージングの結果得られる単語 bigram の品質は，以下に挙げる条件に大きく依存すると考えられる．

1. 抽象化するのは w_i のみか，w_i, w_j の両方か

   単語 bigram のスムージングを行う際には，2つの単語 w_i, w_j ともに抽象化することが望ましい．例えば，w_j が動詞，w_i が動詞 w_j の格要素として現われる名詞である場合を考える．このとき，例えば「食べる」という意味クラス C_j に属する動詞は，「生物」という意味クラス C_i に属する名詞と同じように共起する場合，図 2 の斜線部で囲まれた各事象の確率を等しくすればよい．このような場合は，w_i, w_j ともに意味クラスを用いて抽象化した方が，w_i のみ抽象化するよりもパラメタの数は少ない．

2. シソーラスは木構造かグラフ構造か

   シソーラスが木構造の場合，スムージングに最適な意味クラスの集合の決定は比較的容易に

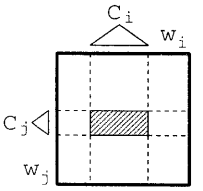

図 2: w_i, w_j ともに抽象化した場合

行える．なぜなら，スムージングに用いる意味クラスの集合を決めれば，各意味クラスに含まれる単語の集合は排他的であるので，単語 bigram の推定は式 (2) により容易に行えるからである．これに対してグラフ構造を持つシソーラスを用いる場合は，スムージングに最適な意味クラスの集合の決定は一般に難かしい．なぜなら，各意味クラスが支配する単語の集合に重なりがあり (図 3(a))，確率モデルは式 (2) のような単純な式では推定できないからである．

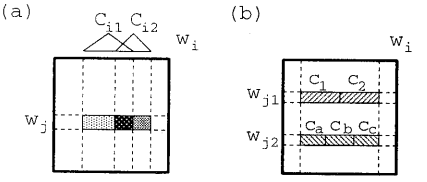

図 3: グラフ構造を持つシソーラスの利用

グラフ構造を持つシソーラスは一般に様々な視点による意味クラスが混在していると考えられ，この性質は単語 bigram の推定に適している．例えば，先ほどの動詞と名詞の単語 bigram について考えると，ある動詞 w_{j1} についてはある視点から見た意味クラス C_1, C_2 を用いてスムージングを行い，ある動詞 w_{j2} については別の視点から見た意味クラス C_a, C_b, C_c を用いてスムージングを行うことも可能となる (図 3 (b))．

3. 意味クラスの抽象化レベルを固定するか否か

今までの研究においては，意味クラスの抽象化レベルを固定してスムージングを行う場合が多かった．しかしながら，先ほど述べたような単語 bigram のスムージングに最適な意味クラスの抽象化レベルは一律に決定できないことが予想される．例えば，動詞と名詞の単語 bigram の場合，ある動詞 w_j について名詞を抽象化するときに，シソーラスの中でも細かい意味クラスを用いるべきところと，粗い意味クラスを用いても構わない部分が混在していることも考えられる．したがって，導入する意味クラスの抽象化レベルを一意に決めてスムージングを行うよりも，導入する意味クラスの抽象化レベルを適宜変更してスムージングを行うことが望ましい．

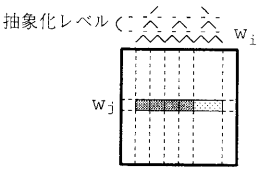

図 4: 様々な抽象化レベルの意味クラスの導入

李らは，抽象化レベルを任意に変えることのできるスムージング方法として，最小記述長原理 (MDL 原理) を応用した手法を提案している [4]．彼らは，動詞とその格要素となる名詞の単語 bigram を推定する際に，スムージングに用いる意味クラスの集合として任意の抽象化レベルの意味クラスを組み合わたもの (彼らは tree cut と呼んでいる) を仮定している．そして，モデルそのものを記述するのに要する記述長 (これはモデルの単純さを表わす) とデータを記述するのに要する記述長 (これは訓練データにどれだけ近いかを表わす) を計算し，その重み和を確率モデルの評価値として，その値が最も小さくなるような意味クラスの集合を求めている．しかしながら，スムージングに用いる意味クラスの集合を決める際にシソーラスが木構造であることを仮定している．また，抽象化レベルを変えるのは格要素となる名詞のみであり，動詞については抽象化レベルを固定している．

今までの考察をまとめると，単語 bigram のスムージング方法としては，(1) 単語 w_i, w_j を同時に抽象化し，(2) グラフ構造を持つシソーラスが利用可能で，(3) 抽象化レベルを適宜変更できることが望ましい．本稿では，最大エントロピー法による確率モデルの推定アルゴリズムを応用すれば，このような単語 bigram のスムージングが行えることに着目した．

3 最大エントロピー法による単語 bigram の推定

最大エントロピー法を自然言語処理に応用した研究としては，N-gram の推定に応用した研究 [3]，品詞付けに応用した研究 [5]，PP attachment 問題に応用した研究 [6]，機械翻訳に応用した研究 [1] などがある．本節では，これらの研究で紹介されている最大エントロピー法による確率モデル推定アルゴリズムについて概説する．このアルゴリズムを単語 bigram に適用する際の問題点とその対処方法については次節にて述べる．

3.1 最大エントロピー法

最大エントロピー法は，訓練データ $C(t,h)$ から，確率モデル $P(t|h)$ を推定するアルゴリズムである．ここで，h は条件付き確率の条件となるべき事象 (以下，履歴事象 (history) と呼ぶ)，t は確率モデルが予測するべき事象 (以下，目標事象 (target) と呼ぶ) であり，$C(t,h)$ は t,h が同時に起こる頻度である．単語 bigram においては，t,h はともに単語であり，h が存在したときにそれが t と共起する確率 $P(t|h)$ を推定することになる．

確率モデル $P(t|h)$ の値は式 (3) で計算される．

$$P(t|h) = \frac{\prod_{i=1}^{|F|} \alpha_i^{f_i(t,h)}}{\sum_t \prod_{i=1}^{|F|} \alpha_i^{f_i(t,h)}} \quad (3)$$

$f_i(t,h)$ は素性 (feature) と呼ばれ，目標事象，履歴事象の組に対して 1 または 0 を返す任意の関数である．また，F はこのような素性の集合である．本稿では素性を (4) のように定義する．

$$f(t,h) = \begin{cases} 1 & t \rightarrow C_t \ \&\ h \rightarrow C_h \\ 0 & \text{その他の場合} \end{cases} \quad (4)$$

C_t, C_h は意味クラスであり，$t \rightarrow C_t$ は単語 t が意味クラス C_t に属することを意味する．例えば，C_t が「生物」という概念を表わす意味クラスで，C_h が「食べる」という概念を表わす意味クラスであるとき，この素性は「生物」を表わす単語と「食べる」という意味を表わす単語が共起しやすいということを表わしている．以下，意味クラス C_t, C_h によって定義される素性を $f_{(C_t,C_h)}$ と記す．このとき，C_t, C_h として任意の抽象化レベルの意味クラス (もしくは単語) を用いることができ，またグラフ構造を持つシソーラスの意味クラスも利用可能である．また，式 (3) 中の α_i は素性 f_i のパラメタである．最大エントロピー法による確率モデル $P(t|h)$ の推定は，これらのパラメタ α_i を推定することに他ならない．

次に，素性の期待値 $E(f_i), \hat{E}(f_i)$ を式 (5), (6) のように定義する．

$$E(f_i) = \sum_{t,h} P(t,h) f_i(t,h)$$
$$= \sum_{t,h} \hat{P}(h) P(t|h) f_i(t,h) \quad (5)$$

$$\hat{E}(f_i) = \sum_{t,h} \hat{P}(t,h) f_i(t,h) \quad (6)$$

但し，$\hat{P}(t,h) = C(t,h) \ / \ \sum_{t,h} C(t,h)$ (7)

$$\hat{P}(h) = \sum_t \hat{P}(t,h) \quad (8)$$

すなわち，$E(f_i)$ とは，f_i が 1 を返す事象の確率モデルにおける確率 $P(t,h)$ の総和 (図 2 では斜線部の確率の総和に相当) である．

最大エントロピー法による確率モデルの推定は，

1. 素性に関する (9) の制約を満たしつつ，

$$P(f_i) = \hat{P}(f_i) \quad \forall f_i \in F \quad (9)$$

2. $P(t|h)$ のエントロピー $H(P)$ が最大となる

$$H(P) = -\sum_{t,h} \hat{P}(h) P(t|h) \log P(t|h) \quad (10)$$

ようにパラメタを推定することにより行われる．すなわち，最大エントロピー法による確率モデルの推定とは，ある素性が 1 を返す事象 (t,h) についてはその確率の和を訓練データのそれに近付け，かつ確率分布が一様分布になるべく近くなるように (意味クラスが支配する単語の確率が全て等しくなるように) 確率モデルを推定することである．

3.2 Generative Iterative Scaling アルゴリズム

Generative Iterative Scaling アルゴリズム (以下，GIS アルゴリズムと呼ぶ) は，与えられた素性の集合 F と訓練データから，式 (9) の制約を満たしながら，かつエントロピーが最大となる確率モデル $P(t|h)$，すなわち式 (3) 中の α_i を決定するアルゴリズムである．この概略を以下に示す．

1. $\hat{E}(f_i)$ を計算する．
2. 全てのパラメタの初期値 $\alpha_i^{(0)}$ を 1 とする．
3. 式 (3) より，与えられたパラメータ $\alpha_i^{(n)}$ における $P(t|h)$ を計算する．
4. 式 (5) より，$E^{(n)}(f_i)$ を計算する．
5. α_i を以下のように更新する．

$$\alpha_i^{(n+1)} = \alpha_i^{(n)} \left[\frac{\hat{P}(f_i)}{P^{(n)}(f_i)} \right]^{\frac{1}{C}} \quad (11)$$

ここで，C は補完定数と呼ばれる定数である．

6. α_i が収束するまで，3.〜5. を繰り返す

GIS アルゴリズムの 1 回の反復に要する計算量は $O(|T||H||F|)$ である．但し，T は目標事象の集合，H は履歴事象の集合である．

3.3 素性選択アルゴリズム

素性選択アルゴリズムとは，与えられた素性の候補の集合 S から，確率モデルに採用するべき素性の集合 F を選択するアルゴリズムである．これは，S

の中から確率モデルのログ尤度 (log likelihood) を最も増大させる素性を 1 つ選択し，それを F に追加するといった操作を繰り返すことによって行う．ログ尤度とは，式 (12) で定義され，エントロピーと同じく確率モデルが一様分布にどれだけ近いか (ログ尤度が低くなればなるほど確率モデルは一様分布に近付く) を表わす指標となるものである．

$$L(P) = \sum_{t,h} \tilde{P}(t,h) \log P(t|h) \qquad (12)$$

この素性選択アルゴリズムは MDL 原理 [4] による確率モデル推定とほぼ一致する．素性を 1 個増やすことは MDL におけるモデル記述長を増やすことに対応し，そのときのログ尤度は MDL におけるデータ記述長に対応する．したがって，素性を 1 個増やしたときのログ尤度の増分が最大となる素性を選択することは，MDL 原理におけるモデル記述長とデータ記述長の重み和が最小となる確率モデルを選択することに等しい．

以下に具体的なアルゴリズムを示す．

1. F を空集合とする．
2. S の各要素 f_j について，f_j を F に加えたときの f_j のパラメタ α_j を求める．本来なら GIS アルゴリズムによって，$\{\alpha_i\}, \alpha_j$ の推定を行うべきであるが (α_i は F 中の素性に対応したパラメタ)，これは計算量が大きく効率が悪い．ここでは，$\{\alpha_i\}$ を変化させず，α_j のみを変動させるように修正した GIS アルゴリズムを適用することによって α_j の値を近似推定した．
3. f_j を F に加えた際のログ尤度の増分 $\Delta L(f_j)$ を計算する．$\Delta L(f_j)$ の最も大きい素性を 1 個選択し，それを確率モデルの素性として新たに F に追加する．
4. 新しい F における各素性のパラメタ α_i を GIS アルゴリズムを用いて推定する．
5. 素性を加えたときの確率モデルのエントロピーの変化量がある閾値 T_{ent} 以下になるまで 2.〜4. を繰り返す．

S の中から素性を 1 個選択するのに必要な計算量は $O(|T||H||S|)$ である．

4 最大エントロピー法による推定の高速化

前節では，最大エントロピー法による確率モデルの推定アルゴリズムについて説明した．ところが，このアルゴリズムを単語 bigram の推定に応用する際には，その計算量が問題となる．GIS アルゴリズムの 1 回の反復に要する計算量は $O(|T||H||F|)$ であり，素性選択アルゴリズムの計算量は $O(|T||H||S|)$ である．単語 bigram を推定するには，T や H が単語の集合となるので，推定に要する計算量は非常に大きく，最大エントロピー法をそのまま適用するのは現実的ではない．そこで本節では，最大エントロピー法による確率モデル推定を効率化するいくつかの手法を提案する．

4.1 GIS アルゴリズムの効率化

- ある履歴事象 h について 1 を返す素性，すなわち (4) における C_h が単語 h を支配する素性の集合を F_h とする．ある履歴事象 h_1 と h_2 において，F_{h_1} と F_{h_2} が全く同一であるなら，式 (3) より，$P(t|h_1)$ と $P(t|h_2)$ は全く同じ確率分布になる．

$$\forall t \in T \quad P(t|h_1) = P(t|h_2) \qquad (13)$$

  したがって，3.2 節の Step 3. の $P(t|h)$ の計算において，F_h が等しい履歴事象については，その中の 1 つの履歴事象 h_1 について $P(t|h_1)$ を計算すれば，その他の履歴事象については計算を省略できる．
- ある事象 (t_1,h) と (t_2,h) において，これらに対して 1 を返す素性の集合が全く同じならば，(14) に示すように，式 (3) の分母の項は全く同じ値になるはずである．

$$\prod_{i=1}^{|F|} \alpha_i^{f_i(t_1,h)} = \prod_{i=1}^{|F|} \alpha_i^{f_i(t_2,h)} \qquad (14)$$

  したがって，3.2 節の Step 3. の $P(t|h)$ の計算において，先ほどと同様に同じ値となる項の計算を省略することができる．

4.2 素性選択アルゴリズムの効率化

- 3.3 節の素性選択アルゴリズムにおいて，集合 S の各素性 f_j について $\Delta L(f_j)$ を計算し，その値が最も高い素性を f_1, 2 番目に高い素性を f_2 とする．3.3 節の素性選択アルゴリズムにおいては，まず f_1 のみを F に加えて確率モデルの推定をやり直してから，再び S 中の素性のログ尤度の増分 (これを $\Delta L'(f_j)$ とする) を計算する．

  ここで，f_1, f_2 が条件 (15) を満たす場合を考える．(このとき，2 つの素性が互いに独立であると定義する．)

$$T_{f_1} \cap T_{f_2} = \emptyset \text{ and } H_{f_1} \cap H_{f_2} = \emptyset \qquad (15)$$

  T_f (または H_f) は素性 f が 1 を返す単語の部分集合，すなわち式 (4) における C_t (または C_h)

が支配する単語 t (または h) の集合である．f_1 と f_2 が互いに独立，すなわち全く異なる事象 (t,h) に対する確率分布を決定する素性である場合は，$\Delta L(f_2)$ と $\Delta L'(f_2)$ はほぼ等しいと考えられる．したがって，f_1 の次に追加される素性として f_2 が選ばれる可能性が高い．このような場合には f_1, f_2 を同時に追加してもよいと考えられる．そこで，素性選択アルゴリズムにおいて，F に素性を 1 個ずつ追加する代わりに，ログ尤度の増分が大きく，かつ互いに独立である上位 N_f 個の素性を F に追加することにした．これにより，素性選択に要する時間を $1/N_f$ に短縮できる．

- 3.3 節のα_jのみ推定するように修正した GIS アルゴリズムにおいては，3.2 節の Step 3. の $P(t|h)$ の計算は以下のようになる．

$$P(t|h) = \frac{\prod_{i=1}^{|F|} \alpha_i^{f_i(t,h)} \times \alpha_j^{(n) f_j(t,h)}}{\sum_t \prod_{i=1}^{|F|} \alpha_i^{f_i(t,h)} \times \alpha_j^{(n) f_j(t,h)}} = \frac{\prod_{i=1}^{|F|} \alpha_i^{f_i(t,h)} \times \alpha_j^{(n)}}{D(\bar{T}_{f_j}) + D(T_{f_j}) \times \alpha_j^{(n)}} \quad (16)$$

$$D(T_{f_j}) = \sum_{t \in T_{f_j}} \prod_{i=1}^{|F|} \alpha_i^{f_i(t,h)} \quad (17)$$

$$D(\bar{T}_{f_j}) = \sum_{t \in \bar{T}_{f_j}} \prod_{i=1}^{|F|} \alpha_i^{f_i(t,h)} \quad (18)$$

ここで，$\bar{T}_{f_j}$ は f_j が 1 を返さない目標事象の集合であり，$T_{f_j} \cup \bar{T}_{f_j} = T$ である．$D(T_{f_j})$, $D(\bar{T}_{f_j})$ の値はα_jの値に全く依存しないので，これらはα_jの推定の第 1 回目の反復時にのみ計算すれば，第 2 回目以降の反復時には計算を省略できる．

- S 中の 2 つの素性の候補 f_1 と f_2 について，それらが 1 を返す目標事象の集合が等しい場合 $(T_{f_1} = T_{f_2})$ を考える．このとき，f_1, f_2 のパラメタ推定において，式 (16) の項 $D(T_{f_j})$, $D(\bar{T}_{f_j})$ の値は全く等しい．したがって，ΔL を計算する素性の候補の順序を T_{f_j} によってソートすることにより，一部の素性の候補については項 $D(T_{f_j})$, $D(\bar{T}_{f_j})$ の計算を完全に省略することができる．

4.3 素性候補選別アルゴリズム

素性の候補の集合の大きさ $|S|$ は素性選択アルゴリズムの計算量に大きく影響する．しかしながら，S の中から確率モデルの学習に無効な素性を事前に削除することができれば，素性選択アルゴリズムの計算量を大幅に削減できる．本節では，S の中から有効な素性を選別する方法を提案する．

素性の候補 $f = f_{(C_t, C_h)}$ が確率モデルの推定にどれだけ有効であるかを表す尺度として，素性の効用 $U(f)$ を以下のように定義する．

$$U(f) = \max\left(U_T(f), U_H(f) \right) \quad (19)$$

$$U_T(f) = \max_{f^T} H(P_{\{f^T\}}) - H(P_{\{f,f^T\}}) \quad (20)$$

$$U_H(f) = \max_{f^H} H(P_{\{f^H\}}) - H(P_{\{f,f^H\}}) \quad (21)$$

ここで，f^T, f^H はそれぞれ f の目標事象，履歴事象について，シソーラスにおける抽象化レベルを 1 段階上げた意味クラスによる素性を表わす．

$$f^T = f_{(C'_t, C_h)} \quad (C'_t \text{は} C_t \text{の上位クラス}) \quad (22)$$

$$f^H = f_{(C_t, C'_h)} \quad (C'_h \text{は} C_h \text{の上位クラス}) \quad (23)$$

また，$P_{\{f,\cdots\}}$ は素性の集合 F が $\{f,\cdots\}$ であるときの確率モデルを表わす．もし，f^T (または f^H) が単語 bigram の推定に最適な素性であるとき，すなわち図 5(a),(b) の斜線部内の事象 (t,h) の訓練データにおける確率がほぼ等しい場合には，これらより抽象化レベルの低い素性 f を F に追加しても，斜線部内の確率分布はほとんど変化しない．

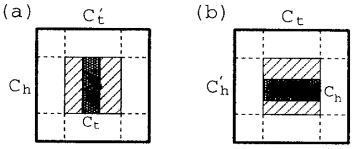

図 5: 素性の効用の計算

このとき，$U(f)$ の値は 0 に近い値を取るはずである．このように，素性の効用 $U(f)$ の値が小さい素性は確率モデルの学習に無効な素性であると考えることができる．

次に，$U(f)$ の計算方法の概略について説明する．まず，条件付き確率モデルのエントロピーを式 (24) のように変形する[1]．

$$H_{(T|h)} \stackrel{def}{=} -\sum_{t \in T} P(t|h) \log P(t|h) = P(T_a|h) H_{(T_a|h)} + P(T_{\bar{a}}|h) H_{(T_{\bar{a}}|h)} + H'_{(T_a, T_{\bar{a}})} \quad (24)$$

$$P(T_a|h) = \sum_{t \in T_a} P(t|h) \quad (25)$$

$$H_{(T_a|h)} = -\sum_{t \in T_a} P(t|T_a, h) \log P(t|T_a, h) \quad (26)$$

$$H'_{(T_a, T_{\bar{a}})} = \sum_{x \in \{T_a, T_{\bar{a}}\}} -P(x|h) \log P(x|h) \quad (27)$$

[1] $P(T_{\bar{a}}|h)$ の定義は (25)，$H(T_{\bar{a}}|h)$ の定義は (26) と同様．

$T_a, T_{\bar{a}}$ は訓練事象の集合 T を二分したときの部分集合である．この式より，T_a の内部の確率分布が変動したとき (図 5 において f を確率モデルに追加したとき) の確率モデル全体のエントロピー $H_{(T|h)}$ の変化は，T_a の中だけの局所的なエントロピー $H_{(T_a|h)}$ の変化と $P(T_a|h)$ との積によって求めることができる．このことから，$U(f)$ の値は素性選択アルゴリズムよりもはるかに少ない計算量で求めることができる．したがって，S 中の各素性の効用 $U(f)$ を計算し，効用の低い素性を事前に削除することにより，素性選択アルゴリズムの計算量を大幅に抑制することができる．

4.4 分割学習

4 節の冒頭で述べたように，最大エントロピー法によって単語 bigram を推定する場合，$|T|, |H|$ の値が大きいことが推定に要する計算量を増大させている原因の 1 つであった．しかしながら，最も抽象化レベルの高い素性 $f_{(C'_t,C'_h)}$ (C'_t, C'_h は最も抽象化レベルの高い意味クラス) が 1 を返す事象 (t,h) の集合が互いに排他的であるのなら，これらの事象の確率分布 $P(t,h)$ は，他の事象とは独立に推定される．したがって，訓練事象 (t,h) をいくつかの集合に分割し，1 つの集合内の事象の確率 $P(t,h)$ が他の集合と独立に推定できる場合，分割された訓練事象のみにおいて素性の選択，および確率モデルの推定を行えばよい．分割された訓練データにおいては，T, H, S の要素数は訓練データ全体に比べて大きく減少するので，確率モデルの推定に要する計算量を大幅に抑制することができる．

4.5 実装

3.2 節の GIS アルゴリズム，および 3.3 節の素性選択アルゴリズムを計算機上に実装した．また，4.1, 4.2 節に述べた手法も取り入れ計算の高速化も試みた．4.3 節の手法についても実装は完了しており，現在その正当性を検証している段階である．実装したプログラムは，単語 bigram の推定だけではなく，一般の確率モデル $P(t|h)$ の推定も行えるように工夫した．現在，以下のような入力に対し，素性選択アルゴリズムを実行して素性の集合 F を決定し，また F から GIS アルゴリズムにより確率モデル $P(t|h)$ を推定することができる．

- 事象 (t,h) の頻度
- 素性の候補の集合 S
  ただし，各素性は次のような形式で記述する[2]．
  1 を返す目標事象のリスト $\{t_1, \cdots, t_n\}$
  1 を返す履歴事象のリスト $\{h_1, \cdots, h_m\}$

このプログラムの使用を希望する方は著者まで連絡されたい．

[2]このとき，この素性は $n \times m$ 個の事象 (t_i, h_j), $(1 \leq i \leq n, 1 \leq j \leq m)$ についてのみ 1 を返す関数である．

5 予備実験

本節では，今まで述べた手法の有効性を確認するために，最大エントロピー法によって実際に確率モデルを学習し，それを実際の複合名詞の係り受け解析に適用した予備実験について説明する．本来なら単語 bigram を推定する実験を行いたいところだが，4 節で計算量を抑制するいくつかの方法を提案したものの，現時点で実装しているのは 4.1 節，4.2 節で提案した手法のみである．そのため，最大エントロピー法を適用して単語 bigram の推定を行えるほど計算量が抑制されたわけではなく，また学習データの量も十分にない．そこで，単語 bigram の推定を行う代わりに，複合名詞中の各単語を意味クラス (分類語彙表における上位 5 桁の分類コード) に変換し，これら意味クラス間の係り受け関係を最大エントロピー法により学習した．

次に，本実験における複合名詞の係り受け関係の解析方法について説明する．例えば，3 つの名詞からなる複合名詞の場合，その係り受け構造には図 6 の (a), (b) 2 つの場合が考えられる．

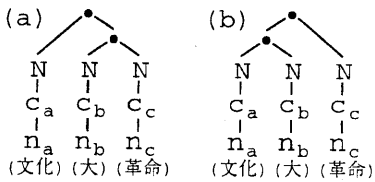

図 6: 複合名詞の係り受け構造

図 (a) の場合，c_a は c_c に，c_b は c_c に係る．このとき，係り受けの候補のスコアを (28) のような確率モデルで計算する．

$$P(c_c|N) \times P(c_b|N[c_c]) \times P(c_a|N[c_c]) \times P(n_a|c_a) \times P(n_b|c_b) \times P(n_c|c_c) \quad (28)$$

ここで $P(n|c)$ は意味クラス c から単語 n に展開される確率，$P(c|N)$ は品詞 N から意味クラス c に展開される確率，$P(c_t|N[c_h])$ は，受け側に意味クラス c_h の名詞があったときに，品詞 N から意味クラス c_t に展開される確率を表わす．図 6 の (b) の場合や，複合名詞に含まれる名詞の数が 4 以上のときも同様に計算する．

式 (28) における確率モデル $P(c_t|N[c_h])$ の推定方法として，次の 4 通りの方法を比較した．

1. 最大エントロピー法で推定した場合 (ME)
   推定の具体的な手順について説明する．まず，頻度 $C(c_t, c_h)$ が5以上の共起データから，c_t, c_h を様々な抽象化レベルに抽象化した意味クラス C_t, C_h を組み合わせた素性 $f_{(C_t,C_h)}$ を作成し，113,711 個の素性の候補の集合 S を作成した．これから，素性選択アルゴリズムにより，N_f を 10(4.2 節)，T_{ent} を 0.01(3.3 節) として 664 個の素性を選択した．さらに，この素性集合 F の下での確率モデルを GIS アルゴリズムにより推定した．
2. 最尤推定した場合 (ML5)

$$P(c_t|c_h) = \frac{C(c_t, c_h)}{\sum_{c_t} C(c_t, c_h)} \qquad (29)$$

3. c_t, c_h を分類語彙表の分類コードの上位 4 桁の意味クラスに変換してから最尤推定した場合 (ML4)
   これは，抽象化レベルを固定したスムージングに対応する．
4. 3. で分類語彙表の分類コードの上位 3 桁の意味クラスを用いた場合 (ML3)

次に，5 文字漢字列 843 個を用意し，これらから得られる全ての漢字分割 (分割数最小のもののみ) とその係り受け構造に対して，上記の各確率モデル (ME,ML5,ML4,ML3) によってスコアを計算した．結果を表 1 に示す．

表 1: 複合名詞の解析結果

	ME	ML5	ML4	ML3
正解	541	274	355	526
不正解	216	44	81	240
同点	3	2	2	12
失敗	83	523	405	65
Rec.(%)	64.2	32.5	42.1	62.4
Pre.(%)	71.5	86.2	81.4	68.7

表 1 中の「正解」はスコア 1 位の候補が正解だった複合名詞の数,「同点」は全ての候補に等しい確率を与えてしまい正解がわからなかった複合名詞の数,「失敗」は未知語などによりスコア付けに失敗した複合名詞の数,「不正解」はそれ以外の複合名詞の数である．また「Rec.」は再現率 (正解数/全体数),「Pre.」は適合率 (正解数/(正解数 + 不正解数)) を表わす．

抽象化レベルを固定したスムージング (ML4,ML3) と最大エントロピー法によるスムージング (ME) を比べると，再現率では ME の方が優れている．適合率を見ると ML4 の方が ME よりも優れているが，解析に失敗した名詞の数は ML4 の方が多く，ML4 ではスムージングが十分に行われていないと考えられる．この結果，最大エントロピー法によるスムージングは抽象化レベルを固定したスムージングよりも有効であると結論できる．

6 おわりに

本稿では，グラフ構造のシソーラスの利用や意味クラスの抽象化レベルを自由に変えることなどを可能とした単語 bigram のスムージング方法として，最大エントロピー法の適用を試みた．また，単語 bigram の推定が可能となるように，最大エントロピー法による確率モデル推定に必要な計算量を抑えるいくつかの手法を提案し，その実装を行った．

今後の課題としては，まず 4.3, 4.4節で提案した方法を実装するなどしてさらなる計算の効率化を図り，実際に単語 bigram の推定に応用することが挙げられる．さらに，複合名詞の係り受け関係だけではなく，動詞とその格要素間の確率モデルや，動詞がどのような格を持つのかを予測する確率モデルなどの学習に適用し，自然言語解析の曖昧性解消に応用していきたいと考えている．

参考文献

[1] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, Vol. 22, No. 1, pp. 39–71, 1996.

[2] Ido Dagan, Fernando Pereira, and Lillian Lee. Similarity-based estimation of word cooccurrence probabilities. In *In Proceedings of Association for Computational Linguistics*, 1994.

[3] 江原暉将. 最大エントロピー法を用いて n グラム確率をバイグラム確率で補完する方法. 言語処理学会第 2 会年次大会発表論文集, pp. 369–372, 1996.

[4] Hang Li and Naoki Abe. Generalizing case frames using a thesaurus and the mdl principle. In *In the Proceedings of Recent Advances in Natural Language Processing*, pp. 239–248, 1995.

[5] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *In Proceeding of the Empirical Methods in Natural Language Processing Conference*, 1996.

[6] Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. A maximum entropy model for prepositional phrase attachment. In *In Proceeding of the Human Language Technology*, pp. 250–255, 1994.