# 空間分割型 PLSI を用いた言語横断情報検索

**高野 祐介** † **村松 哲** † **森 辰則** ††
† 横浜国立大学 大学院 環境情報学府 †† 横浜国立大学 大学院 環境情報研究院
E-mail: {yusukeb,tetsu0730,mori}@forest.dnj.ynu.ac.jp

## 1 はじめに

近年，インターネットの普及により，ユーザーは，電子文書を容易に入手できるようになり，世界中の多くの情報にアクセスすることができる．しかし，欲しい情報がユーザーの母国語で書かれたものであるとは限らない．したがって，ユーザーが母国語をキーワードとして，関連する他の言語の文書をも検索できるような言語横断情報検索（CLIR) の要求が高まっている．検索要求の言語と検索対象文書の言語が異なる場合，そのままでは，検索ができない．その代表的な方法として，検索要求の言語を検索対象の言語に翻訳する方法である検索要求翻訳型言語横断検索がある．その際に単語を翻訳するのだが，一つの単語に対して複数の訳語候補がある場合，どの訳語が適当であるかを判断する基準は少ない．これを，訳語の曖昧性といい，一般的に短い文書に対して翻訳を行う場合に曖昧性は大きくなる．この曖昧性を減少する手法に翻訳候補に順位をつけるものがあり，その一つに，予め大量の対訳文書から，翻訳確率情報を集める手法がある．

本稿では，確率的 LSI(PLSI) に基づいて，質問翻訳による言語横断情報検索を行う方法を提案する．まず，対訳コーパスに対して PLSI を適用することにより，個々の語に対してその訳語の共起確率 (翻訳確率) を計算できることを示す．PLSI は，行列計算なので対訳コーパスが大規模になると時間/空間計算量が非常に大きくなり一般の計算機では処理できなくなる．そこで，対訳コーパスを分野に応じて小さく分割し，個々に PLSI を計算し，得られた翻訳確率を合成することにより最終的な翻訳確率を得る方法を検討する．

## 2 PLSI(Probabilistic Latent Semantic Indexing)

PLSI[1] は、自動インデクシングの手法の１つである．まず，各文書に現れる単語頻度を求め，単語ー文書行列を作る．各列が文書に対応するベクトルとなっており，その各成分は各単語に対応する．ここで各単語についてのベクトルと見ることで，単語の特徴量とすることができる．しかし，単語-文書行列は一般に疎であるためなんらかのスムージングが必要である．PLSI では，非観測変数を用いて，これに対応している．単語と文書の条件付確率 $P(w|d)$ を非観測変数 $z \in Z$ を用いて，$P(w|d) = \sum_{z \in Z} P(w|z)P(d|z)P(z)$ とすることができる．単語の条件付生起確率 $P(w|z)$, 文書の条件付生起確率 $P(d|z)$ は，非観測変数 $z$ の次元に圧縮される．

これを，CLIR に応用させるためには，検索要求翻訳の際に，対訳辞書内の翻訳候補に優先度を持たせるための翻訳情報として用いることが考えられる．本稿では，PLSI に基づく質問翻訳型 CLIR を提案する．

まず，対訳文書対を１つの文書と見立て，その文書に出現する単語の頻度を求め，単語-文書行列を得る．この単語行列に対して PLSI を行い単語の条件付確率を得る．対訳の単語同士は，文書内集合での分布が似てくると考えられ，ある日本語単語 $w_j$ が英語単語 $w_e$ に翻訳される確率 $P(w_e|w_j)$ は，PLSI で求められた各単語の条件付生起確率 $P(w|z)$ を用いて，次式のように求められると考える．

$$P(w_e|w_j) = \frac{P(w_e, w_j)}{P(w_j)} = \frac{\sum_{z \in Z} P(w_e|z)P(w_j|z)P(z)}{\sum_{z \in Z} P(w_j|z)P(z)} \quad (1)$$

## 3 空間分割型 PLSI

PLSI を大規模コーパスにおいて用いる場合，前述のように単語の条件付確率を得ようとする時，単語-文書行列が大きくなるほどその処理にかかる計算量が増大し，一般の計算機では扱えなくなってしまう．

そこで本稿では，コーパスを分割して各分割コーパスに対して PLSI を行い，それによって得られた各翻訳確率を合成することで，最終的な翻訳確率として扱う手法を提案する．

### 3.1 空間分割

対訳コーパスを，いくつかの部分コーパスに分割することを考える．この時，各部分コーパスから得られる確率情報は，１つの分野 (domain) における翻訳情報を表現していると考えられる．本稿では，コーパス分割により得られた分野の各々を空間と呼ぶ．そして，各空間から得られた翻訳情報は次のように表せる．

$$\begin{aligned} P(w|z) &\rightarrow P(w|z, dom) \\ P(d|z) &\rightarrow P(d|z, dom) \end{aligned}$$

$P(z)$ は，分野に依存せずに決定される．

### 3.2 分割空間と検索要求の関係

各分割コーパスから得られた翻訳確率を合成するために，各空間と，検索要求との関係を導く．

まず，空間集合 $DOM$ 内のある１つの空間 $dom$ は，文書集合であり，検索要求 $Q$ は，分野に依らない１つの文書であるとする．$Q$ が与えられた時にある空

間 $dom$ が選択される確率 $P(dom|Q)$ は次式で計算される．

$$P(dom|Q) = \frac{P(Q|dom)P(dom)}{\sum_{dom' \in DOM} P(Q|dom')P(dom')} \quad (2)$$

$$= \frac{P(Q|dom)}{\sum_{dom' \in DOM} P(Q|dom')} \quad (3)$$

$$P(Q|dom) \cong P(q_1 \dots q_m|dom) \quad (4)$$

$$\cong \prod_{q_i \in Q} P(q_i|dom) \quad (5)$$

$$P(q_i|dom) = \frac{\sum_{d \in dom} n(d, q_i)}{\sum_{d \in dom} \sum_{w' \in d} n(d, w')} \quad (6)$$

ただし，$Q$ 内の単語を $q_1 \dots q_m$ とし,$P(dom)$ は，空間 $dom$ に依らず一定であるとする．

ここで，各空間での頻度 $n(d, q_i)$ が 0 となる単語が存在する場合，$P(dom|Q)$ 値が 0 となってしまい，その空間の情報が無視されてしまう．しかし，コーパスの表現する標本空間で 0 であっても実際の確率が 0 でないこともある．本稿では，そのようなスパースネス問題を解決するために，グッド・チューリングの推定 [2] を用いて，スムージングを行った．

### 3.3 翻訳確率の合成

各空間の翻訳確率を考えると，次のように表せる．

$$P(w_e|w_j, dom) = \frac{\sum_{z \in Z} P(w_j|z, dom)P(w_e|z, dom)P(z)}{\sum_{z \in Z} P(w_j|z, dom)P(z)} \quad (7)$$

これを全ての空間にわたって合成すると次式のようになる．

$$P(w_e|w_j) = \sum_{dom \in DOM} P(w_e|w_j, dom)P(dom|Q) \quad (8)$$

この合成によって得られた翻訳確率を用いて，訳語候補における優先度を決定する．この決定法を用いることで，検索要求によって空間選択の確率が変わることにより，検索要求の文脈にそった訳語の決定が合わせてできると考える．

## 4 評価実験

実験には NTCIR2 テストコレクションを用いて行った．

### 4.1 対訳辞書の作成

本稿では，NTCIR の日英対訳コーパスにより，対訳コーパス内で各文書のキーワード部分が，単語単位でほぼ対訳になっている (表 1) ことに着目し，この対応を利用して対訳辞書を作成した．

このとき,「トリーイング」は「treeing」,「画像処理」は「image prossing」に翻訳されるような辞書を作成する．対訳コーパス内のキーワードについてこの手順で処理を行い，それを辞書とした．キーワードを取り出す際には，stemming[3] 処理を行った．また，誤字に対しては，Levenstein 距離を用いて誤字の修正を行った．

### 4.2 翻訳確率の作成

NTCIR2 プロジェクトでは，日英対訳文書 339483 件が使用可能である．しかし，全てのタグの項目について対訳にはなっておらず，使用に際して有効性があると思われる，タイトル，キーワード，アブストラクトが対訳となっている 180802 件を抽出し，これをトレーニングコーパスとした．

このトレーニングコーパスを，文書中の学会タグに基づいて 10 の分野に分割した．また，コーパスの大きさによる比較のために各々の分野において 20 % (35883 件) の量の対訳コーパスも用意した．分割に対しての比較として，20，30，40 分割のものを，用意し，非観測変数の数を 100，200，500 として翻訳率を作成した．

### 4.3 文書のインデクシング

検索課題及び検索対象文書から，予め日英対訳文書から作成した対訳辞書に含まれる単語，つまりキーワードに対して頻度を調べ，インデクシングをした．

日本語について処理する単位としては，本来 chasen や juman 等のツールを用いるのだが，本稿では対訳辞書にある被翻訳語と翻訳語についての情報を得なければならず，単に単語単位でのインデクシングは適切でない．従って，対訳辞書つまり対訳文書内のキーワードの形を単位としてインデクシングを行った．英語についても同様に辞書作成時の処理を行い，辞書内の形を単位としてインデクシングを行った．

### 4.4 検索方法

検索トピックの中から，多くの語を検索に用いるために <TITLE>, <DESCRIPTION>, <NARRATIVE> を用いて，検索要求語とした．その検索要求に対して，翻訳確率によって翻訳した．そして，ベクトル空間法を用いて検索を行った．

### 4.5 比較実験

本システムの翻訳部分での性能を比較するために人手によって翻訳した検索要求での検索，トレーニングコーパスを用いる手法として分割を用いない PLSI での翻訳での検索，単語間の相互情報量による共起判定に基づく翻訳での検索，$\chi^2$ 分布による共起判定に基づく翻訳での検索の 4 手法を比較実験として行った．

人手での翻訳に関しては NTCIR2 の日本語，英語の検索要求はお互いに翻訳の関係にあるのでそれを用いた．この評価は，検索精度の上限を，計るものとして位置付けている．本手法での実験では，検索システム自体に直接関るものではないので，検索システムとして用いたベクトル空間法の検索の精度を表していると考える．

表 1: 対訳コーパスでのキーワードの対応例

| |
|---|
| <KYWDTYPE = ”kannji”> トリーイング // 画像処理 // ポリエチレン </KYWD> |
| <KYWETYPE = ”alpha”> treeing // image prossing // polyethylene </KYWE> |

$$\chi^2 = \frac{N(N_{++}N_{--} - N_{+-}N_{-+})^2}{(N_{++} + N_{+-})(N_{-+} + N_{--})(N_{++} + N_{-+})(N_{+-} + N_{--})} \tag{9}$$

単語間の相互情報量は次式を用いた.

$$I(w_j, w_e) = log_2 \frac{P(w_j, w_e)}{P(w_j)P(w_e)} \tag{10}$$

$$= log_2 \frac{P(w_e|w_j)}{P(w_e)} \tag{11}$$

$$= log_2 \frac{P(w_j|w_e)}{P(w_j)} \tag{12}$$

$$P(w) = \frac{\sum_{d \in D} n(d, w)}{\sum_{d \in D} \sum_{w' \in d} n(d, w')} \tag{13}$$

$w_j$ は，日本語単語，$w_e$ は，英語単語を示す．また $P(w_e, w_j)$ は，日本語と英語が，同一文書内にある確率を示し，$P(w_j), P(w_e)$ は，各々の単語の出現確率を表している.

$\chi^2$ 分布による共起判定には式 (9) を用いた [4]. $N_{++}$ は，日英両方の単語が出現した単語数．$N_{+-}$ と $N_{+-}$ は，どちらか片方だけ出現した単語数，$N_{--}$ は両方とも出現しなかった単語数を示す.

## 5 結果

結果として図 1 に示す．日英，英日は翻訳の方向を示す．項目の「同一言語」は、人手での翻訳での検索結果である．「CL-LSI」は，本手法と同様にコーパスを分割する手法を用いている CL-LSI[5] での結果であり，参考として示す．ただし，条件 [1] は必ずしも一致していない.

### 5.1 PLSI を用いた結果

他の手法と比べ，本手法である空間分割を用いた PLSI は，日英，英日ともに各条件で高くなった．しかし，同一言語での結果と大きく差が開いている．各条件においての $z$ の変化については，ほぼ同じような結果となった.

### 5.2 コーパスの違い，分割の違いによる結果

英日での結果では,「10 分割」において，100%コーパスを用いたものが 20% を用いたものより低くなった．20 分割以上の値で比べると，高くなることが示せた.

[1] 検索要求内の TITLE タグを用いておらず，単語の抽出方法として C-value[6] に基づく手法を用いている

分割に関しては，日英，英日において，分割が大きくなるほど精度が若干高くなる傾向が見られるが，はっきりとした相関が見られる程ではなかった.

### 5.3 DISCRIPTON のみでの結果

検索質問トピックのかなで,DISCRIPTON タグをのみを用いた実験も行ったが，すべての条件において低い値になった．これは,DISCRIPTON タグの中の単語数が少ないことが考えられる．本手法では, 単語をインデクシングする際に, コーパスから作成した対訳辞書内の単語によって行なうので検索要求が短いと正しく翻訳される語の絶対数が少なくり，検索精度が低下すると考える.

## 6 考察

### 6.1 PLSI による翻訳精度について

非分割での結果のみを考えると，他の手法と比べ同等かそれ以上の精度であることがわかる．分割手法を用いることで，さらに精度が向上することが確認できた.

### 6.2 大量コーパスを用ての翻訳精度について

大量コーパスからの翻訳確率生成では，10 分割での 20%と 100%の比較より 20%の結果のほうが高くなった．しかし，20 分割以上のものは，精度が良くなった．これは，1 分割あたりの文書数が関係していると考えられる．つまり，各分割文書集合から翻訳確率を生成する際に，単語数が多くなるため，確率分布が均一になってしまう．それにより翻訳語の決定において確率の差がなくなり精度が落ちると考えられる.

以上のことより，分割数を考慮に入れることを前堤として，分割したコーパスを合成する本稿の手法で，大量コーパスを用いることでの精度の向上が確認できた．分割手法を用いている CL-LSI との比較においても日英，英日共に本手法の精度が良いこが確認できた.

また，1 ドメインあたりの文書数を考えたとき，非分割 (35883 文書) に対して各分割での文書数は表 2 のようになっている．コーパスの 20%を使用した非分割手法での空間計算量に対して，分割手法では，コーパス 100%を 10 分割した場合でも，半分程度であることが分かる．したがって，大量コーパスを低い空間計算量で扱えることを確認できた.

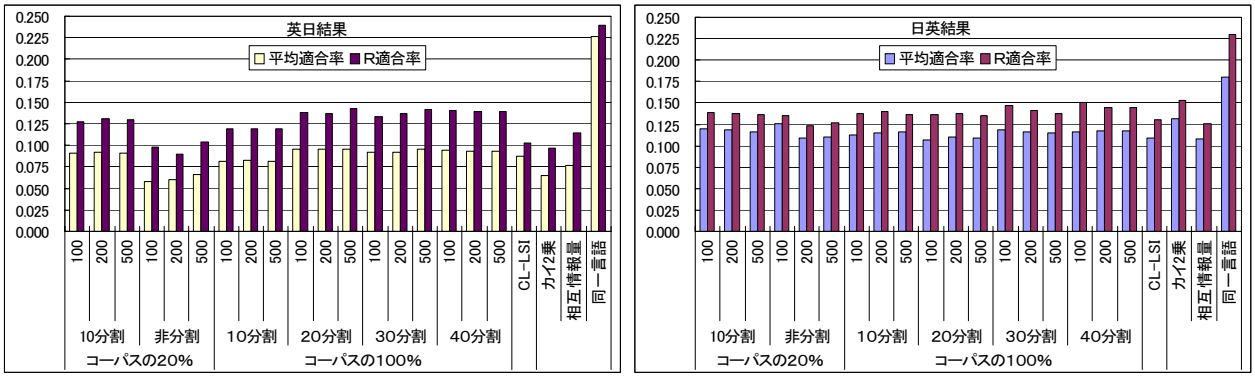


図 1: 左 英日結果 右 日英結果

図 2: 1 ドメインあたりの平均文書数

| 分割数 | 10 | 20 | 30 | 40 | 10(20%) |
|---|---|---|---|---|---|
| 平均文書数 | 18080.2 | 9040.1 | 6026.7 | 4520.05 | 3568.6 |

項目の 10(20%) はサイズが 20%での 10 分割である

### 6.3 分割による翻訳精度について

20%コーパスにおける,「10 分割」と,「非分割」を比較することで分割手法が非分割手法よりも効果的なことが分かる．これについては，次のようなことが考えられる． まず，非分割の手法では，検索要求と翻訳確率との係わりは無いので，訳語撰択に際してコーパス依存性が高い．しかし，本手法では，検索要求に特化するように重みづけをするのでコーパス依存が少なくてすむ．これにより検索に有用な訳語決定ができたと考えられる．

また，分割数を考えると分割数が大きい程若干の精度の向上が見られた．これは，分割数が大きいと，翻訳確率を求める空間が小さくなるのでかたよりをもった確率が生成される．つまりその空間での，よりはっきりとした翻訳確率ができる．そして，合成の際に，検索要求に基づく重みづけによって，どの空間を採用すれば良いかという事がきまる．この結果より全体をみての確率よりも局地的に大きい確率をうまく採用するほうが，訳語の決定に有効に働くと考えられる．このように，分割手法を用いることで，PLSI の翻訳精度が高くなることが確認できた．

## 7 おわりに

本稿では言語横断検索において，検索要求語の翻訳の際に，PLSI に基づき複数に分割した対訳文書から求めた翻訳確率を検索要求に基づいて合成し，合成した翻訳確率を用いて対訳候補から翻訳語を決定することを試みた．

PLSI による翻訳候補の決定では，非分割手法では，コーパス依存が大きくなり，比較実験での結果とさほど変わらなかった．しかし，分割手法での，検索における訳語選択の有効性を確認できた． また，大量のコーパスを用いた際には，１分割あたりの文書数を考慮に入れることにより，翻訳精度の向上することが確認された．しかし，人手での翻訳との比較では，まだ開きがある．これについては，翻訳辞書内の訳語候補に人手で翻訳された語が無い場合や，キーワードの取り方により人手での翻訳よりも，翻訳対象語が少ない場合などがあり，精度が落ちると考えられる．

以下の点を，今後の課題としたい．

上記に記してある，対訳辞書，インデクシングの問題である．本手法では，対訳辞書，インデクシングキーワードを対訳コーパスによって得た．そのことによって，単語の誤字，検索有効単語取りこぼしなどが考えられる．これについては，一般の対訳辞書等での本手法の応用が考えられるが，膨大な量の単語へのインデクシングが必要となり，この部分での工夫が不可欠である．

## 参考文献

[1] Thomas Hofmann. Probabilistic Latent Semantic Indexing. *SIGIR'99*, pp. 50–57, 1999.

[2] 北研二, 中村哲, 永田昌明. 音声言語処理 (コーパスに基づくアプローチ). 森北出版株式会社, 1996.

[3] Recardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM press, 1999.

[4] Hinrich Schutze. Automatic word sense discrimination. *Computaional Linguistics,Volume24 Number1,Association for Computaional Linguistics*, 1998.

[5] 森辰則, 国分智晴, 田中崇. 空間分割型 CL-LSI による大規模言語横断情報検索. 情報処理学会論文：データベース，vol.43 No.SIG 2(TOD 13), pp. 27–36, 3 月 2002.

[6] K. Frantzi and S Anaiadou. Extracting nested collection. *Proc. 16th International Conference on Computational Linguistic (COLING 96)*, pp. 41–46, 1996.