空間分割型 CL-LSI による大規模言語横断情報検索

森 辰則[†] 國分 智晴[††] 田中 崇[†††]

本稿では，Latent Semantic Indexing（LSI）方式による言語横断情報検索において問題となる，大規模な対訳コーパスの利用方法について考察する．大規模対訳コーパスを用いて単語空間を作成しようとすると，LSI の要である単語–文書頻度行列の特異値分解が記憶装置の制約で難しくなるとともに，語の訳の曖昧性が非常に大きくなるという問題がある．そこで，文書の類似度に従って，対訳コーパスを適切な複数の部分対訳コーパスに分割し，各々の単語空間を作成する手法を提案する．この方法では，検索対象の文書を，最も類似した部分対訳コーパスから構成された単語空間に配置することによって，訳語の曖昧性を減少させる．検索時には，検索質問をそれぞれの単語空間に配置し，文書ベクトルとの類似度計算を行う．このときに，単語空間ごとの未知語に対する重み付けの補正が重要であり，検索精度が 10%〜20%程度向上することを示す．

Large-scaled Cross-Language Information Retrieval Based on Segmented CL-LSI

Tatsunori Mori,[†] Tomoharu Kokubu[††] and Takashi Tanaka[†††]

In this paper, we report the utilization of a large-scaled bilingual corpus in Cross-Language Latent Semantic Indexing (CL-LSI). When we construct one monolithic word space with a large-scaled corpus, we face the problems such as the increase of ambiguity in the translation of words, the difficulty in the Singular Value Decomposition, which is the essential process in LSI. In order to cope with the problems, we introduce the method in which the large bilingual corpus is divided into smaller sub-corpora according to the similarities among documents. Each of sub-corpora yields one word sub-space. By placing each document in one of the word sub-spaces, which is the most similar sub-corpus to the document, the ambiguity of translation is expected to be decreased. In the retrieval of documents, queries are placed in all of word sub-spaces, and similarities between the queries and the documents are calculated. We show that the adjustment in the similarity calculation for unknown words is very helpful to increase the effectiveness in retrieval.

1. はじめに

近年，インターネットの発展などにより外国語文書を電子的に入手する機会が急激に増えており言語の壁を越えた情報検索技術である言語横断検索（CLIR）の要求が高まってきている[12]．言語横断検索において現在主流の方法は何らかの形で対訳辞書を用いている．そこでは辞書作りの過程で人間が対訳情報を吟味しているので，翻訳に関する精度が良いと考えられる．その精度は対訳辞書の質や規模のみならず，その使用方法に依存するところが大きい．

一方，対訳コーパスなどの言語資源から対訳情報を自動的に抽出し，言語横断検索に利用する研究がある．これらの手法は，対訳辞書を用いずに言語横断検索が可能であるという点で魅力的であり，ある程度の精度で検索ができることが報告されている．その手法としては，対訳辞書を自動生成し，検索質問の翻訳に役立てるといった手法や，ベクトル空間法などにおいて文書の中間表現を作成する際に利用する方法などがある．後者の手法として代表的なものが，Cross-Language Latent Semantic Indexing（CL-LSI）である．Carbonell ら[3] によると，中規模コーパス（1134 対訳）による実験では，事例に基づく機械翻訳による検索質問翻訳手法が最も性能が良く，CL-LSI 手法と GVSM 手法[1] がこれに次ぎ，最も性能の悪かった手法が一般

† 横浜国立大学大学院環境情報研究院社会環境と情報部門
Division of Social Environment and Information Studies, Graduate School of Environment and Information Sciences, Yokohama National University
†† 株式会社東芝研究開発センター知識メディアラボラトリー
Knowledge Media Laboratory, Corporate Research and Development Center, TOSHIBA Corporation
††† 横浜国立大学大学院工学研究科電子情報工学専攻
Division of Electrical and Computer Engineering, Graduate School of Engineering, Yokohama National University

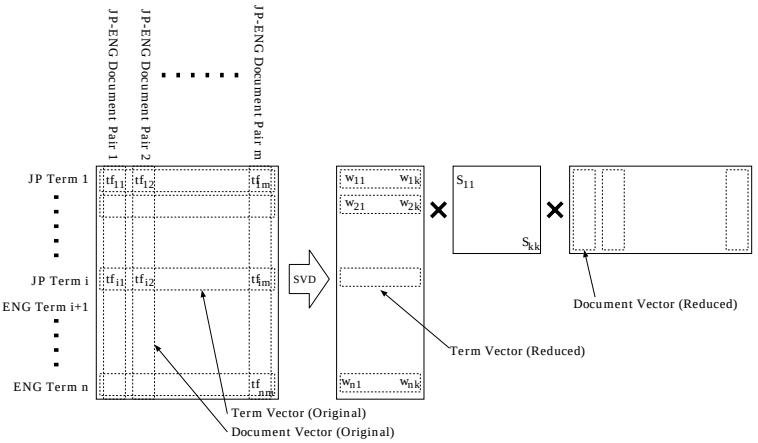

図 1 言語横断型 LSI
Fig. 1 Cross-Language Latent Semantic Indexing.

対訳辞書を用いた検索質問翻訳手法であった．

しかし，大規模な文書集合を検索対象とする場合に，CL-LSI などの手法が，どのような性能になるかはいまだ不明である．特に，CL-LSI は，大規模対訳コーパスに対処できないという問題がある．すなわち，大規模文書集合を検索対象とするためには，必然的に同等の範囲・規模を持つ大規模対訳コーパスを用いて，翻訳に関する情報を抽出しなければならない．一方，CL-LSI 方式では単語空間を作成する際に大規模な対訳コーパスを用いると，LSI の要である単語–文書頻度行列の特異値分解が記憶装置の制約で難しくなる．さらには，複数の分野にまたがっての頻度累計を行うため，訳語の曖昧性が非常に大きくなる．

そこで，本稿では，対訳コーパスを文書の類似度に従って複数の部分対訳コーパスに分割し，各々の単語空間を作成する手法を提案する．この方法では，検索対象の文書は，最も類似した部分対訳コーパスから構成された単語空間に配置されるので，訳語の曖昧性が減少する．検索時には，検索質問をそれぞれの単語空間に配置し，文書ベクトルとの類似度計算を行うことにより検索を行う．このときに，単語空間ごとの未知語に対する重み付けの補正が重要であることを示す．

2. Cross-Language Latent Semantic Indexing

Cross-Language Latent Semantic Indexing（CL-LSI）は，検索質問の翻訳を必要としない，完全自動の言語横断検索手法である[5]．これは複数の言語を含む「意味」空間を Latent Semantic Indexing（LSI）により自動的に生成することによりなされる．

2.1 Latent Semantic Indexing

LSI の特徴は，単語を次元とする典型的なベクトル空間法ではなしえない，語と語の間の関係を自動的にモデル化し，検索効率を向上させることにある[4]．言語横断検索においては，日本語単語と英単語のように，表層表現がまったく異なる語の間の関係を見いださなければならないので，LSI の持つその特徴が非常に重要になる．LSI では，まず，単語–文書頻度行列を作成する（**図 1** 左側）．この行列の要素 (i, j) は文書 j における語 i の頻度である．行列の各行は，ある単語が文書中にどのように出現したかを表す情報であるから，その単語の出現する文脈を表現すると考えられる．これら文脈から単語間の重要な連想関係を発見するために，線形代数の手法である特異値分解（Singular Value Decomposition）が単語–文書頻度行列に適用される．これにより，似通った文脈に登場する語が近くに配置されるように次元を縮退した特徴量空間が形成される．これを LSI 空間と呼ぶ．通常のベクトル空間法では各語は文書ベクトルの各次元に対応するので，互いに直行するベクトルとして表現される．一方，LSI においては，語は文脈を表す縮退された単語ベクトル（以下,「縮退単語ベクトル」と呼ぶ）によって表現されるので，必ずしも，語に対応するベクトルの間に線形独立性はない．2 つの語が似通った文脈で使われている場合には，LSI 空間において類似したベクトルとして表現される．

新しい文書や検索質問文は，それを構成する語に対応する縮退単語ベクトルの重み付き線形和により，LSI 空間に畳み込まれる．以下では，便宜上，畳み込みにより得られた文書に対応するベクトルを「畳み込み文書ベクトル」と呼ぶ．なお，文脈によって誤解の生じ

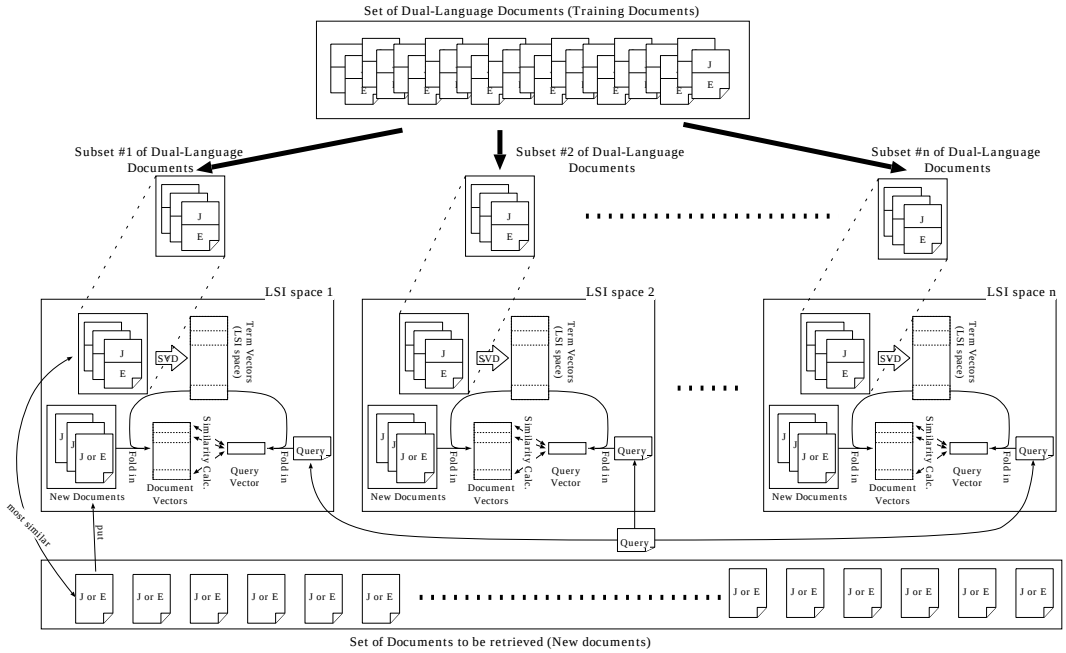

図 2 空間分割型 CL-LSI の枠組
Fig. 2 Scheme of segmented CL-LSI.

ない場合は単に「文書ベクトル」と記すこともある．

検索は，検索質問文に対応するベクトルと文書に対応するベクトルとの間の類似度を cosine 相関度などにより計算し，順位付けを行うことで達せられる．

2.2 LSI による言語横断検索

LSI は簡単に言語横断検索に適用できる．CL-LSI は対訳文書集合から縮退単語ベクトル，すなわち，LSI 空間を作成する訓練段階と，(一般に対訳ではない) 検索対象文書を LSI 空間に配置し検索を行う段階に分かれる．訓練段階においては，対訳文書対を 1 つの文書に見たて，その文書に出現する単語の頻度を求め，単語–文書頻度行列を得る．この行列に対して LSI と同様に SVD を適用し，縮退単語ベクトルを得る（図 1 右側）．互いに訳語となる単語どうしでは，文書対における出現の分布が似ているので，LSI 空間上の近い位置に配置されることが期待される．この性質により，言語を横断することが可能となる．対訳になっていない検索対象文書は，LSI における新文書の扱いと同様であり，構成単語に対応する縮退単語ベクトルにより LSI 空間に畳み込む．

3. 大規模コーパスにおける CL-LSI の問題点

CL-LSI 方式は文書の扱う対象領域がある程度限定されている場合に有効な手法である．対象領域が広範囲にわたる場合，LSI 空間を作成するにあたって，語の網羅性を高めるように対訳文書対を集めるとすると，文書対の数が大きくなる．これは SVD において計算量の問題を生じさせる．SVD は行列操作であるので，行列の次元が高くなれば，それに応じて記憶資源を消費する．よって，語彙サイズを大きくするために非常に大きな対訳コーパスを使おうとすると，空間計算量において破綻する．たとえば，NTCIR-1 [10] で公開された論文要旨の対訳コーパスにおいては，約 18 万対訳があり，単語数は複合語を含めて約 37 万語であった．これを行列の要素数にすると，67×10^9 要素であるから，0 要素が多く含まれていたとしても，中規模程度の計算機の主記憶装置に入れることは不可能である．

そこで我々は，訓練のために用いるコーパスを計算機の資源にあわせて分割し，各々の部分コーパスから別々な LSI 空間を生成する方法を提案する．その枠組みを**図 2** に示す．コーパスの分割により，まず，SVD が可能となり，さらに，各部分コーパスごとに分野がある程度限定されていれば，訳語の曖昧性の減少に役に立つと期待される．しかしながら，訓練コーパスを分割するにあたって，少なくとも，1) どのようにコーパスを部分コーパスに分割するか（4.1 節），2) 複数の LSI 空間に対して，いかに検索対象文書を配置するか（4.2 節），3) 配置された検索対象文書をどのように検索するか（4.3 節），を明らかにする必要がある．

次章では，これら検討事項に注意しつつ，我々の枠組みについて述べる．

4. 大規模対訳コーパス向け CL-LSI

4.1 LSI 空間の分割

部分コーパスにおいて，その中の文書の分野が限定されていると，文脈も自ずと限定され，個々の単語における対訳の多様性も軽減されると考えられる．よって，複数の LSI 空間を構築するにあたっては，類似度に従って対訳文書を複数の部分グループに分割することが有効であろう．これを自動的に行う手法としては，各種クラスタリングアルゴリズムが知られている．しかし大規模な対訳文書群に対してクラスタリングを行うには，多大な計算機資源が必要とされる．また，文書部分集合の分割数については，利用可能な計算機資源によるので最終的な調整は人間の手によるところが大きい．

そこで，別の手法を検討する．実際の文書には分野を表す情報が付加されていることも多い．たとえば学術論文を考えると，個々の文書に学会名などの分野名に関する情報が付与されている．この情報を利用し，本稿では以下に述べる半自動的なクラスタリングを行う．すなわち，同じ分野名を持つ文書対を 1 つのグループと考え，グループを併合・分割し，適切な大きさのグループを作成する．なお，後に述べる実験では学会名を分野名として利用した．

(1) 対訳文書を分野名によって分類し，分野グループを作成する．
(2) 各対訳文書の文書ベクトルを作成する．このベクトルは語を次元とし要素を対応する語の tf·idf 値とする．
(3) 同一分野グループ内の文書ベクトルの平均を求め，それを分野ベクトルとする．
(4) 文書数の多い分野グループを数個，手作業で選択し，主要分野グループとする．
(5) 残りの分野グループの各々について，最も類似度の高い主要分野グループに併合する．類似度は，分野ベクトルの間の方向余弦により求める．
(6) 文書数が上限を超えた分野グループは，上限を満たすように分割する．文書数の上限は計算機資源により決める．
(7) 各分野グループについて分野グループ内の文書ベクトルの平均を求め分野ベクトルを更新する．

4.2 検索対象文書の配置

本手法においては LSI 空間は分野グループによって異なるので，どの LSI 空間に文書を配置するかによって，異なる（畳み込み）文書ベクトルが作成される．複数の LSI 空間がある状況において，文書ベクトルを作成する方法には主に，検索対象文書をすべての LSI 空間に配置する方法と，1 つの LSI 空間を選択し，そこに配置する方法が考えられる．前者の方がすべての翻訳情報を利用するので，検索の精度が良いと考えられるが，LSI 空間の数だけ個別の文書ベクトルが必要であり，記憶装置もその分占有してしまう．これは特に大規模文書データベースを作成するときに問題となるので，本稿では選択した 1 つの LSI 空間のみに配置する．

この方式では，検索対象文書を配置する LSI 空間を適切に選択しなければならない．ある検索対象文書の配置先は，訳語選択の分野依存性から，同一の分野の対訳から作成された LSI 空間であることが望ましい．よって，各検索対象文書を，最も類似する分野ベクトルを持つ分野の LSI 空間に配置する．検索対象文書を LSI 空間に畳み込む方法としては次式を用いる☆．

$$\mathbf{D} = \sum_{T_i \in D} tf(T_i, D) idf(T_i) \mathbf{T}_i \qquad (1)$$

$\mathbf{D}$: 検索対象文書 D の（畳み込み）ベクトル
$tf(T_i, D)$: D 中の語 T_i の頻度
$idf(T_i)$: T_i の idf 値 $\log \frac{N}{df(T_i)} + 1$. $df(T_i)$ は T_i の文書頻度. N は総文書数
$\mathbf{T}_i$: T_i のベクトル

4.3 複数 LSI 空間での文書検索

CL-LSI 方式では検索質問も他の検索対象文書と同様に LSI 空間上のベクトルとして表現される．検索質問に対する各検索対象文書の順位付けは，検索質問ベクトルとの間の類似度に基づきなされる．しかし，本手法では LSI 空間が複数あるので，次の手順で文書検索を行う．

(1) 検索質問をすべての検索対象文書と比較するために，検索質問ベクトルを各 LSI 空間に 1 つずつ作成する．
(2) 各 LSI 空間ごとに，検索質問ベクトルとすべての（畳み込み）文書ベクトルの間の類似度を cosine 相関度により計算する．
(3) その類似度を複数の LSI 空間にわたって，降順に整列し，検索対象文書に対する順位付けを行う．

なお，ここでは最も素朴かつ基本的な方法として，異なる LSI 空間，すなわち異なる基底による空間での

☆ idf 値を導入している点で，Dumais らの手法[5] とは異なる．

類似度をそのまま併合している点に注意されたい．後に記す実験により実証的な面よりこの併合の有効性の検討は行うが，理論的にはさらなる考察が必要である．たとえば，異なる空間間の類似度を併合するにあたっては，空間間の関係を何らかの方法で調べ，類似度の尺度の変換を行うことが考えられる．具体的には，形成されたすべての LSI 空間において既知である語のみを持つ参照文書を数点，各空間に配置し，その各々と検索質問との間の類似度を計算することにより，各空間の類似度の値の較正を行うことがその候補である．この検討は今後の課題としたい．

5. 対訳コーパスの分割による未知語の問題

CL-LSI 方式では対訳コーパス中に現れない単語については，その対訳情報が得られないので，文書ベクトルを作成するときに無視される．よって，検索対象文書中には現れるが，対訳コーパスに現れない語が検索質問中に含まれるときに，検索精度が低下する．これは原理上不可避である．

一方，我々の方式には，これとは異なる種類の未知語の問題がある．コーパスを分野ごとに分割した場合，関連する部分コーパスのみに現れ，他の部分コーパスに現れない語が存在しうる．このため，LSI 空間ごとに未知語が異なる可能性があるので，文書検索において期待どおりの結果が得られないことがある．

例として，T_a，T_b，T_c という 3 つの語からなる検索質問 $\mathbf{Q}(T_a, T_b, T_c)$ で LSI 空間 TS_1 と TS_2 中の文書を検索する場合を考える．LSI 空間 TS_1 には，語 T_a が存在し，T_b，T_c が未知語となっているとする．そこには，T_a のみが含まれる検索対象文書 $\mathbf{D}_1(T_a)$ が配置されているとする．また，LSI 空間 TS_2 には，語 T_a，T_b，T_c のすべてが存在するとし，ここに，T_a，T_b が含まれる検索対象文書 $\mathbf{D}_2(T_a, T_b)$ を配置するとする．

ここでは検索質問に含まれる語との一致の度合いから，$\mathbf{D}_1(T_a)$ よりも $\mathbf{D}_2(T_a, T_b)$ の検索順位を高くしたく，$\mathbf{Q}$ と $\mathbf{D}_2$ の間の類似度の方が $\mathbf{D}_1$ の場合よりも大きくなることを期待する．しかし，上述の状況においては，逆で，$\mathbf{D}_1$ の類似度の方が $\mathbf{D}_2$ よりも大きくなってしまう．これは TS_1 での類似度計算が，T_a のみについて行われるので，検索質問があたかも T_a であると見なされるためである．一方，TS_2 での類似度計算は，T_a，T_b，T_c について行われるので，検索質問と文書 $\mathbf{D}_2$ の間の類似度は $\mathbf{D}_2$ が T_c を持たない分，低くなる．

この例が示すとおり，我々の望む類似度計算を行うためには，検索質問中の未知語を単に無視するのではまずく，類似度を低下させる要因として適切に扱わなければならない．その方法の 1 つとして，コーパスを分割する際に，すべての部分コーパスにおいて，必ずすべての語が現れるように制御する方法が考えられる．つまり，まず，語の網羅性を保証する共有部分コーパスを作成し，次に，残りのコーパスを分野に応じて分割し，共有部分コーパスに加えて，各部分コーパスを作成する．しかし，索引の対象となる語のほとんどは文書頻度がさほど大きくないことから，共有部分コーパスの規模を低く抑えることは難しく，各部分コーパスの大きさを計算機資源に応じた適切な値に制御することもできない．さらに，部分コーパスの分野も制御できないという問題もある．

そこで，我々は，コーパスの分割の方法に依存しない手法として，任意の未知語に対応する新しい次元を 1 つ各 LSI 空間に導入する方法を提案する．この方法は，次の手順で LSI 空間を拡張し，未知語を既存のどの縮退単語ベクトルとも直行するベクトルとして扱うことにより，類似度に対する補正をする．

ある LSI 空間が k 次元空間で表現されているとすると，単語は $(w_1, \ldots, w_k)$ なるベクトルになる．この空間に対して，新たな次元を導入し (k+1) 次元とする．このとき，既存の単語については，$(w_1, \ldots, w_k, 0)$ とし，一方，検索質問に現れるすべての未知語を，$(0, \ldots, 0, 1)$ とする．（畳み込み）文書ベクトルにおいて，(k+1) 次元目の成分がつねに 0 であることを考慮すれば，補正の前後で以下の各式が成り立つ．

$$\mathbf{D}' \cdot \mathbf{Q}' = \mathbf{D} \cdot \mathbf{Q} \tag{2}$$

$$|\mathbf{D}'| = |\mathbf{D}| \tag{3}$$

$$|\mathbf{Q}'| = |\mathbf{Q} + \mathbf{Q}_u| = \sqrt{|\mathbf{Q}|^2 + \left(\sum_{T_i \in Q_u} tf(T_i, Q_u) idf(T_i) \right)^2} \tag{4}$$

$\mathbf{D}$，$\mathbf{D}'$：補正前後の畳み込み文書ベクトル

$\mathbf{Q}$，$\mathbf{Q}'$：補正前後の検索質問ベクトル

Q_u：検索質問中の未知語のリスト

$\mathbf{Q}_u$：Q_u に対応する部分の検索質問ベクトル

よって，補正後の文書ベクトル $\mathbf{D}'$ と検索質問ベクトル $\mathbf{Q}'$ の間の類似度 $sim(\mathbf{D}', \mathbf{Q}')$ は，cosine 相関度を用いるとすると次式となり，新たに構築された空間での類似度は，検索質問に含まれる未知語の分だけ低く見積もられる．

$$sim(\mathbf{D}', \mathbf{Q}') = \frac{\mathbf{D}' \cdot \mathbf{Q}'}{|\mathbf{D}'||\mathbf{Q}'|}$$

$$= \frac{\mathbf{D}\cdot\mathbf{Q}}{|\mathbf{D}|\sqrt{|\mathbf{Q}|^2+\left(\sum_{T_i\in Q_u} tf(T_i,Q_u)idf(T_i)\right)^2}} \quad (5)$$

6. 評価実験

6.1 索引語の認定

索引語には単語ならびに複合語を用いた．日本語文書については，形態素解析器 JUMAN 3.61 により，単語切り出しならびに品詞付与を行った．品詞情報により，名詞，形容詞，動詞，連体詞，副詞，アルファベット，カタカナを索引語として取り出した．英語については，Porter 法[7] で語基抽出（stemming）を行うとともに，Fox[6] に基づきストップワードリストによる不用語の削除を行った．

複合語の認定には，日英両言語で一貫して扱える手法として，C value に基づく方法を用い，次の 2 段階で行った[8]．まず，対訳コーパスより，語を単位とするサフィックスアレイを作成し，単語列の出現頻度を求めた．そしてある閾値 TH_f 以上の出現回数を持つ単語列を複合語の候補とした．次に各複合語候補に対して，C value を計算し，その値がある閾値 TH_c 以上のものを複合語として認定した．今回の実験では，プログラムの制約上，NTCIR-1 の対訳コーパスを 11 に分割し，各部分集合において，$TH_f = 5$，$TH_c = 5$ の条件の下で，上記手続きを行った．なお，複合語の構成要素も索引付けに用いている．

6.2 LSI 空間の分割に関する実験

LSI 空間を複数に分割して作成し検索を行う我々の方式では，対訳コーパスの情報を一度に参照していないという点で，単一 LSI 空間を用いた方式より精度が劣る可能性がある．一方で，分野ごとに LSI 部分空間を作成すれば，分野に応じて対訳情報が得られる可能性があるので，精度が向上する可能性もある．そこで，この点を確認するために，手元の計算機環境で単一空間を構成できる範囲で，メイト検索☆による評価実験を行った．

まず NTCIR-1 の言語横断タスクから得られた学会情報付きの対訳技術文書（要旨）6000 対を訓練コーパスとして用いた．検索対象としては訓練コーパスとは別の対訳文書（3000 対）を用いた．これについて，次の 3 つの場合を比較した．

☆ 検索対象文書集合から文書を 1 つ選択し，その対訳を検索質問とする．このとき，元の文書が何位で検索されるかを見ることにより，検索精度を検証する手法[5]．

表 1 空間分割型 CL-LSI の精度評価
Table 1 Effectiveness of segmented CL-LSI.

LSI 空間	1 位 (%)	3 位以内 (%)
全体	58.2	75.7
分割 (学会)	47.8	63.9
分割 (学会) 補正後	59.4	78.2
分割 (均等)	47.4	67.8
分割 (均等) 補正後	47.0	67.4

(1) 訓練コーパスを分割せずに 1 つの LSI 空間を作成・検索を行った場合（**表 1** 中の「全体」）
(2) 訓練コーパスを学会情報を基に 3 等分し，部分 LSI 空間を作成，検索対象文書を提案手法により検索した場合（表 1 中の「分割（学会）」）
(3) 文書の所属する学会が均等に混在するように訓練コーパスを 3 等分する以外は上記 2 と同じ場合（表 1 中の「分割（均等）」）

いずれの場合も，各対訳集合から 2 章で述べた方法により，単語ベクトルの集合を得た．各単語ベクトルの次元は SVD により 154 次元〜159 次元に縮退された．SVD には SVDPACKC[2] を用いた☆☆．

結果を表 1 に示す．ここで,「1 位」とは対訳文書対の一方を検索質問として検索したときに，その対訳文書が順位 1 位で得られたことを表す．また,「3 位以内」とは対訳文書が 3 位以内に得られたことを表す．

6.3 NTCIR-2 における実験

大規模文書集合に対する実験として，NTCIR-2 の言語横断タスクにおいて評価を行った[10]．その検索規模は非常に大きく，提案手法が大規模な検索タスクでどの程度の精度となるのかを評価できる．訓練コーパスとしては NTCIR-1 で使用された日英技術文書要旨約 38 万件を使用することが可能で，本実験ではそのうち日英の対訳対が得られた約 18 万文書対を訓練用に使用した．これは，57 学会の論文要旨の対訳対からなる．検索対象文書は日英技術文書要旨約 70 万件であった．ここで，NTCIR-2 の言語横断タスクにおいては，NTCIR-2 で新たに加わった文書に加えて，NTCIR-1 で用いられた文書も検索対象文書となっていることに注意されたい．たとえば，英語文書についていうとその内訳は**表 2** に示すとおりであり，半数以上は訓練コーパスに由来するものである．

まずは，我々の最初の目標である，空間分割型 CL-LSI の大規模コーパスへの適用可能性を確認した．上

☆☆ 縮退後の次元数は訓練コーパスから構築された単語–文書頻度行列のランクに依存している．この実験では SVD の過程における繰返し数（Lanczos step）を 400 に指定しているが，一方で，得られた次元はそれより小さい 150 次元程度であるので，行列のランク数により適切な縮退次元数が決まったと考えられる．

表 2 英語検索対象文書の由来
Table 2 Origin of English documents to be retrieved.

	文書数	比率
NTCIR-1 由来の英語文書	187,080	58.1 %
NTCIR-2 で新たに加わった英語文書	134,978	41.9 %

記訓練コーパスを先に述べた手順で部分コーパスに分割した．具体的には，最初に要旨対数の最も多い 6 つの学会を選択し独立したクラスタとした．次いで，残りの学会の文書対を最も類似度の高いクラスタに配置した．計算機資源を考慮して，そのうちの 4 クラスタを 2 つに分割し，最終的に 10 の文書集合を得た．各集合の大きさは約 14,000 から約 26,000 文書対であった．また，単語の種類数は約 78,000 から約 115,000 であり，全体では約 380,000 語であった．この分割の結果，我々の手元にある計算機環境（CPU：UltraSPARC-II 450 MHz，主記憶装置：1 GByte）で 10 個の文書対集合各々に対して，順次 LSI を適用することが可能となり，また，情報検索も可能となった．なお，各単語ベクトルの次元は SVD により，430〜463 次元に縮退された☆．LSI の実行と索引作成のために使用した外部記憶装置は延べ 6 GByte 程度であった．

次に情報検索の精度を評価するために，検索実験を行った．NTCIR-2 には検索トピックとして日英各々49 件あり，我々の実験では，DESCRIPTION フィールド（1 文程度）単体を検索質問とした場合と，DESCRIPTION と NARRATIVE（要約文書程度）を合わせたものを検索質問とする場合を調べた．そして，各検索ごとに類似度の上位 1,000 件を求め，これを NTCIR-2 の適合判定結果と照合することにより，平均適合率ならびに R 適合率を求めた．ここでは関連性評価 S ならびに A のものを適合文書とした．

結果を**表 3** に示す．‘J-E’ は日本語を検索質問とし，英語文献を検索対象とする場合であり，‘E-J’ はその逆である．‘Desc’ は DESCRIPTION フィールドを検索質問とした場合であり，‘Desc-Nar’ はさらに NARRATIVE フィールドを検索質問に加えた場合である．

さて，分割されたどの部分コーパス上にも出現しない語を「完全な未知語」と呼ぶことにする☆☆．本方式では，完全な未知語については縮退単語ベクトルがどの LSI 空間にも存在しないので，その語が検索質問に出現した場合に検索精度が悪くなる．そこで完全な未知語のない検索質問においてどの程度の精度が見込まれるかを別途評価した．結果を**表 4** に示す．

最後に，検索結果中の適合文書の由来について調査した．先に述べたように，検索対象文書の中には訓練に用いた NTCIR-1 に由来する文書が含まれている．本実験ではこれら文書から対訳情報を取得しているため，仮に検索結果中の適合文書において NTCIR-1 に由来する文書が支配的であったとすれば，未知の文書に対する言語横断検索が可能となっているとはいい難い．そこで，最も検索精度が高く，その可能性が高い ‘J-E-Desc-Nar’（表 3）について，検索時と同様に各トピックについて上位 1,000 件を調べ，適合文書（関連性評価 S ならびに A）の由来について調査した．その結果を**表 5** に示す．この表を，先に述べた検索対象となる英語文書の比率（表 2）と比較すると，むしろ，検索結果中の適合文書において NTCIR-1 に由来する

表 3 すべての検索質問に対する平均適合率，R 適合率
Table 3 Average precision and R-precision for all queries.

	Average precision	R-Precision
J-E-Desc	0.0533	0.0635
補正後	0.0666	0.0786
補正による改善	24.9 %	23.8 %
J-E-Desc-Nar	0.0868	0.1031
補正後	0.0940	0.1096
補正による改善	8.3 %	6.3 %
E-J-Desc	0.0512	0.0705
補正後	0.0610	0.0839
補正による改善	19.1 %	19.2 %
E-J-Desc-Nar	0.0609	0.0876
補正後	0.0736	0.1018
補正による改善	20.8 %	16.2 %

表 4 未知語のない検索質問に対する平均適合率，R 適合率
Table 4 Average Precision and R-Precision for queries that do not contain unknown words.

	検索質問数	Average precision	R precision
J-E-Desc	43	0.0600	0.0704
補正後	43	0.0743	0.0870
補正による改善		23.8 %	23.6 %
J-E-Desc-Nar	31	0.1032	0.1206
補正後	31	0.1094	0.1307
補正による改善		6.0 %	8.4 %
E-J-Desc	43	0.0579	0.0782
補正後	43	0.0692	0.0942
補正による改善		19.5 %	20.4 %
E-J-Desc-Nar	39	0.0738	0.1025
補正後	39	0.0872	0.1187
補正による改善		18.1%	15.8 %

☆ 先の実験と同様，縮退後の次元数は訓練コーパスから構築された単語–文書頻度行列のランクに依存している．SVD の過程における繰返し数（Lanczos step）は 1000 に指定している．

☆☆ 5 章で述べた補正の対象となる「対訳コーパスの分割により生ずる未知語」は，ある分割訓練コーパスに注目すると未知語であるが，他のいずれかの分割訓練コーパスには必ず出現している点が，「完全な未知語」と異なる点である．完全な未知語を含まない検索質問においても，分割された訓練コーパスによっては補正の対象となる未知語を持つ場合があるので注意されたい．

表 5 'J-E-Desc-Nar' における適合文書の由来
Table 5 Origin of Relevant Documents.

	文書数	比率
NTCIR-1 由来の適合文書	526	35.3 %
NTCIR-2 で新たに加わった適合文書	962	64.7 %

文書の方が少ないことが分かる．

7. 考　　察

7.1 実験結果に関する考察

1 つめの実験（表 1）によれば，LSI 空間を分割した場合，分割しない場合に比べて，大幅な精度低下が見られる．しかしながら，学会情報に基づいて分割した場合には，未知語の効果を補正すると，分割しない場合と同等もしくは若干の精度の向上が見られる程度まで性能が改善することが分かった．よって，中規模実験ながら，空間分割型 LSI では，未知語の補正が必要であること，そして，単一空間による LSI よりも性能が低くならないことが確認できた．

分割の仕方による差を考えると，まず，未知語の補正の前においては，その検索精度が変わらないことが分かる．これは，コーパスが分野ごとにまとまっていても，そうでなくても，得られた縮退単語ベクトルを用いると，ある程度の言語横断ができることを示唆するものである．一方で，未知語の補正についていえば，コーパスが分野によって分割されている場合にだけ有効であり，コーパスが分野によらず均等に分割されている場合には，その効果がないことが分かった．均等に分割されている場合は，各空間に存在する単語の種類が多くなり，空間ごとの語彙に差がほとんどなくなるために補正が効かないのは容易に予想されることである．これに対して，分野ごとに分割している場合に補正がこれほど有効であることの理由については考察を要する．我々の提案する未知語の補正式（5）について考えると，類似度の補正係数は検索対象文書ではなく，その文書の属する LSI 空間に応じて決定される．つまり，検索質問に含まれる語を網羅する空間に属する文書ほど類似度が高くなり優先される．これは，空間が分野によって分割されている場合には，未知語の補正により間接的に分野選択が行われていることに相当する．この分野選択の機能が検索精度向上に役立っていると考えられる．

2 つめの実験（表 3，表 4）では，次のことが確認された．まず，空間分割型 CL-LSI が 18 万文書対を訓練に利用するような大規模な検索環境においても，適用可能であることが確認された．利用した計算機資源が通常のエンジニアリングワークステーションであることから，一般の計算機環境で実現可能であることが示された．また，表 5 ならびに表 2 が示すように，検索結果は訓練コーパス（NTCIR-1 コーパス）中の文書に偏ることなく，新規文書中の適合文書も検索できている．これは，訓練コーパスに過多に依存することなく言語横断検索ができることを示している．

次に情報検索の精度を考察する．絶対的な性能評価の観点からすると，やはり人手で構築した対訳辞書に基づく手法に比べ，検索精度がかなり低いといわざるをえない．NTCIR-2 における最も良いシステムの平均適合率が 0.3 を超えているのに対し，我々の手法では約 0.1 である．しかし，ある程度の規模の対訳コーパスがあれば，大規模な言語横断検索もある程度の精度で可能であるということが確認された．また，我々の導入した補正手法の効果は大規模コーパスを対象にした場合でも確認される．特にそれは DESCRIPTION フィールドだけを用いた場合の方が，DESCRIPTION ならびに NARRATIVE フィールドを用いた場合よりも顕著である．これは，短い検索要求に未知語が現れたときには，その影響が相対的に大きくなってしまうためである．

一方，コーパスに現れる単語のみからなる検索質問に限定した場合（表 4）に，精度が向上していることから，やはり，完全な未知語については，LSI に基づく方式で不可避の問題として依然として残ることが分かる．

さらに，我々と同様にコーパス情報のみを用いた手法により NTCIR-2 に参加したシステムと比較してみる．Jiang ら[9] は，Approximate Dimension Equalization という手法を導入している．この手法は，より少ない特異ベクトルの計算により，LSI の持つ効果を達成するものである．NTCIR-2 の評価実験においては，NTCIR-1 の対訳文書を学習用に用いた場合，J-E ならびに E-J の平均適合率が，それぞれ，0.0724，0.0829 であることが報告されている．両者の平均は 0.0777 である．この実験においてどのフィールドを検索質問に使用しているかは不明であるが，我々の結果が，J-E，E-J の平均で 0.0638（DESCRIPTION フィールドのみ），0.0838（DESCRIPTION+NARRATIVE）であるから，ほぼ同等の性能と考えられる．

対訳コーパスを用いる両手法において，ほぼ一致したさほど高くない精度しか得られなかったことから，NTCIR-1 コーパスから得た対訳コーパスが NTCIR-2 で新たに加わった検索対象文書に適合していなかったことが考えられる．

一方で，本稿では，空間分割型 CL-LSI の効果を調

べることに注力するために，一番基本的な枠組みで考察を行い，言語横断検索で精度向上のために用いるいくつかの常套の手法を利用していない．たとえば，翻訳過程が大きく関与する初期検索に加えて，そこでの再現率の低さを補うために，疑似フィードバックなどを用いて，繰返し検索を行い精度を向上させるのが通例である．実際に，NTCIR-2 に参加した精度が上位のシステムではフィードバックを多重に適用していた．我々の枠組みを初期検索として用い，適切なフィードバック手法を利用すれば，精度の向上が期待できる．これは，今後の課題としたい．

7.2 計算量に関する考察

1 つめの実験（表 1）が示すとおり，空間の分割と精度はトレードオフの関係にある．補正が正しく効いている場合には，分割前後の精度の差がなくなる傾向にはあるが，計算機資源が許す限り分割数は極力抑える方がよいと考えられる．

空間計算量において，使用する主記憶装置の大きさに関していうと，Lanczos 法を用いた SVD の場合，部分コーパスから得られる単語–文書頻度行列の非零要素の数にほぼ比例する．また，単語–文書頻度行列の非零要素の数は文書数に依存してほぼ決まる．よって，当然ではあるが，分割数を大きくし各分割コーパスに含まれる文書数を少なくすれば，空間計算量がそれだけ減少する．

一方，外部記憶装置に関する空間計算量は，検索対象文書に対応するベクトルの数が変わらないために，単語ベクトルを縮退する次元数が同じであれば，空間を分割する前後で変わりはない．

時間計算量についていえば，空間分割によって影響を受ける可能性があるのは，縮退単語ベクトル作成時の SVD の計算量，検索質問に対するベクトルを作成する計算量，ならびに，検索時の類似度の計算量である．

Lanczos 法を用いた SVD の場合，その時間計算量は各 Lanczos ステップに要する計算量にステップ数を乗じたものにほぼ等しい．各 Lanczos ステップにおける時間計算量を支配するのは単語–文書頻度行列にベクトルを乗じ新たなベクトルを得る計算であり[2]，その計算量は同行列の非零要素にほぼ比例する．一方，空間を分割して得られた行列群の非零要素の数の総計は元の行列のそれに等しい．よって，分割計算における各種オーバヘッドを除けば，基本的には時間計算量は変わらないと考えられる．

一方，検索質問に対するベクトル作成についていえば，1 つの検索質問に対して LSI 空間の数だけベクトル作成を行うため，未知語の差はあるものの，ほぼ，空間数に比例して計算量が増加する．

検索質問と検索文書との類似度計算については，時間計算量の変化はない．なぜならば，SVD の後に得られる縮退単語ベクトルならびにそれから得られる畳み込み文書ベクトルにおいては，零成分がほとんどないので，検索質問に対するベクトルとの類似度計算をすべての文書に対して行う必要があるため，検索時間は検索対象文書数に比例する．よって分割の前後で差がない．

8. おわりに

本稿では，既存の対訳コーパスのみを翻訳の情報として用いる情報検索手法として，CL-LSI 手法に注目した．我々は，これを大規模対訳コーパスに適用するために，訓練用の対訳コーパスを分割し，複数の LSI 空間を併用する方法を提案した．また，LSI 空間ごとに異なる未知語に起因する検索精度低下について検討し，対処方法を提案，その効果をメイト検索により確認した．さらに，NTCIR-2 における評価実験により，大規模文書集合を対象とした検索においても，同手法が適用できることを示し，他の対訳コーパス方式と同等の性能が得られることが確認された．ただし，その性能は対訳コーパスに基づかない他の手法に比べると低い精度にとどまっている．

今後の検討課題としては，先に述べたフィードバック手法や，複数空間における類似度の併合方法についての検討がある．また GVSM [1),3)] などの他の類似方法で大規模コーパスを用いた場合との比較をする必要がある．さらに，最近，文書ベクトルの次元圧縮技術について，いくつかの新しい試みがあるので，それらとの関連を調べることも重要である．たとえば，高速主成分分析手法を用いる黒岩らの研究[13)] や，Non-negative Matrix Factorization を用いる柘植らの方法[11)] がある．いずれも，SVD における計算量の問題を回避する有力な手法として注目される．しかし，コーパスの規模に比べて小さな主記憶装置において計算を行う場合には，やはり，行列の分割などの工夫をし，主記憶装置を有効に利用することが有効であろう．

謝辞 NTCIR を企画・運営し，評価用データを作成していただいた皆様に感謝いたします．なお，本研究の一部は科学研究費補助金基盤研究（C）（2）課題番号 11680383 の補助を受けている．

参 考 文 献

1) Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, ACM Press, Addison Wesley Longman Ltd. (1999).
2) Berry, M., Do, T., O'Brien, G., Krishna, V. and Varadhan, S.: *SVDPACKC* (*Version 1.0*) *User's Guide*, Computer Science Department, University Tennessee (1993).
3) Carbonell, J.G., Yang, Y., Frederking, R.E., Brown, R.D., Geng, Y. and Lee, D.: Translingual Information Retrieval: A Comparative Evaluation, *Proc. International Joint Conference on Artificial Intelligence '97 IJCAI '97* (1997).
4) Deerwester, S., Dumais, S.T. and Harshman, R.: Indexing by Latent Semantic Analysis, *Journal of the Society for Information Science*, Vol.41, No.6, pp.391–407 (1990).
5) Dumais, S.T., Letsche, T.A., Littman, M.L. and Landauer, T.K.: Automatic cross-language retrieval using Latent Semantic Indexing, *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval* (1997).
6) Fox, C.: Lexical Analysis and Stoplists, *Information Retrieval — Data Structure & Algorithms*, Frakes, W.B. and Baeza-Yates, R. (Eds.), chapter 7, pp.102–130, Prentice Hall PTR (1992).
7) Frakes, W.B.: Stemming Algorithms, *Information Retrieval — Data Structure & Algorithms*, Frakes, W.B. and Baeza-Yates, R. (Eds.), chapter 8, pp.131–160, Prentice Hall PTR (1992).
8) Frantzi, K. and Ananiadou, S.: Extracting Nested Collocations, *Proc. 16th International Conference on Computational Linguistics* (*COLING 96*), pp.41–46 (1996).
9) Jiang, F. and Littman, M.L.: Approximate Dimension Reduction at NTCIR, *Proc. NTCIR Workshop 2 Meeting*, pp.5–179–5–74 (2001).
10) NTCIR Project: NTCIR (NII-NACSIS Test Collection for IR Systems) Project Web page, http://research.nii.ac.jp/ntcadm/index-en.html (2000).
11) 柘植 覚, 獅々堀正幹, 北 研二：Non-negative Matrix Factorization を用いた情報検索，自然言語処理研究会 NL-142-1, 情報処理学会 (2001).
12) 菊井玄一郎：言語の壁を越えて文書を検索する—クロスランゲージ情報検索，人工知能学会誌，Vol.15, No.4, pp.550–558 (2000).
13) 黒岩眞吾，柘植 覚，田仁宏典，Xiaoying, T., 獅々堀正幹，北 研二：Simple PCA を用いたベクトル空間情報検索モデルの次元削減，自然言語処理研究会 NL-144-9, 情報処理学会 (2001).

(平成 13 年 9 月 25 日受付)
(平成 13 年 12 月 27 日採録)

(担当編集委員 江口 浩二)

森 辰則（正会員）

昭和 39 年生．平成 3 年横浜国立大学大学院工学研究科博士課程後期修了．工学博士．同年，同大学工学部助手．現在，同大学大学院環境情報研究院社会環境と情報部門助教授．自然言語処理，情報検索，情報抽出等の研究に従事．平成 10 年 2 月〜11 月米 Stanford 大学客員研究員．

國分 智晴

昭和 51 年生．平成 13 年横浜国立大学大学院工学研究科博士課程前期修了．同年（株）東芝入社．現在，同社研究開発センター知識メディアラボラトリー勤務．横浜国立大学大学院在学中は情報検索に関する研究に従事．

田中 崇

昭和 51 年生．平成 12 年横浜国立大学工学部電子情報工学科卒業．現在同大学大学院工学研究科博士課程前期在学中．情報検索に関する研究に従事．