オンライン対訳文書対からのテキスト領域抽出とアラインメント

品川哲也（横浜国立大学大学院環境情報学府）
森辰則（横浜国立大学大学院環境情報研究院）
影浦峡（東京大学大学院教育学研究科）

1　はじめに

インターネットを介した多言語情報流通が盛んになる中、主にオンラインの文書を翻訳し、オンラインで翻訳文書を公開するボランティア翻訳者の活動が活発になっている。我々は現在、特にこれらの翻訳者を念頭に、その支援システムを構築している。

こうした翻訳者のニーズを調べてみると、基本的に望んでいるのは、レファレンス・ツールの内容および検索機能の強化である。言語単位の観点からは、とりわけ熟語・慣用句の検索機能、固有名や専門語のカバー範囲と検索機能、引用句のオンラインでの検索機能、簡単な連語のオンラインでのチェック機能が強く求められている [1]。これらの言語単位のうち、引用句および専門語・固有名について、翻訳者は、独立したレファレンス・ツールがカバーする範囲の拡充を望むと同時に、自分が対象としている分野に関連する、既に訳された文書から関連する情報を引き出してチェックしたいと強く望んでいる。

そこで我々は、指定されたテーマの対訳文書をウェブから収集するシステム AKIN および登録された URL から対訳文書を収集するシステム AKIN2 を構築してきた [2]。これらのシステムは、いずれも対訳ページの URL を出力としており、関連既訳文書をリサイクルし活用するためには、その URL を入力とし、様々な不要情報を削除して対訳文書をアラインし、有用な言語単位の対を抽出する必要がある。

本稿では、その第一段階として、Web 上の不要な情報を削除しつつ、段落単位で対訳文書間をアラインする手法について報告する。

なお、本研究は、既存の対訳アラインメント研究の多くと比べると [3, 4, 5, 6]、乱雑で多様なタグを含む実文書を扱う点で位置づけが異なる。また、Web をパラレル・コーパスとして活用する研究としては、同一サイト内で対応する `index.html` と `index-j.html` などを対象する、データ指向の収集手法があるが [7]、我々が扱うような、目的指向で収集された Web 文書対訳対は、原文書と翻訳文書でまったく異なる構造とまったく異なる不要情報を持っている点に特色がある。

2　基本方針とシステムの構成

本システムは、翻訳者が求めるテーマに応じた様々な文書対を扱うため、ウェブ文書中での HTML タグの使用方法やタグ付与の詳細度等も多様であることが想定される。今のところは英日対訳対のみを扱っているが、様々な言語対への拡張を考えている。これに対応するために、基本的な方針として、できるだけ単純かつ頑健で適用範囲の広い手法を組み合わせて用い、その結果を見ながら徐々に複雑な方法を必要に応じて取り入れ、パフォーマンスを改善してゆく方針をとることとした。

システムの基本概念構成を図 1 に示す。

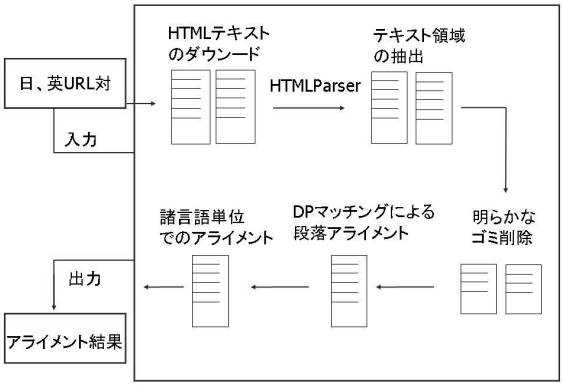

図 1. システムの基本概念構成

日英の対訳文書 URL 対を入力とし、Web サイトから HTML 文書をダウンロードする。これに対して、タグとテキスト領域とを基本的な処理対象として、まず HTMLParser を用いてテキスト領域を抽出し、そこから明らかな不要箇所を削除して、文書領域候補を残す[1]。次いで、タグと文書中の特徴とを使った DP マッチングにより、段落単位でのアラインメントを行う。

以下、手法を順に説明し、そのあとで実験の結果を報告することにする。

3　文書領域候補抽出

まず、HTMLParser を用いて、タグを手がかりに、段落付けを行うと同時にタグを外したテキスト領域を取得する。これに対して以下の処理を行う。

3.1　不要箇所のタイプ

HTML テキストから文書の対訳を取り出そうとする場合、テキスト中に現れる様々な不要部分と文書領域の区別が重要な課題になる。不要領域には、広告情報などの明らかな不要領域と、翻訳者の意図する注釈・解説等の文書に関係する不要領域（文書不要領域）が存

[1]本稿では、「文書領域」という言葉を対訳関係にある文書そのものの領域、「テキスト領域」という言葉を HTML 文書のタグを外した部分を指すために用いる。

在する。これら２つの不要領域の違いとして段落を構成する文字数があげられる。一般に明らかな不要領域は段落を構成する文字数が少なく、文書不要領域は文字数が多くなる。

HTML テキストの開始と終わりには明らかな不要領域が塊として存在することが多く、不要領域の多いテキストでは文書領域が数十段落であるのに対し、不要領域の段落が数百存在するものもある。多くは文書領域の前後に塊として存在するものである。

文書領域候補抽出モジュールでは、完全に文書領域を認定することを目標とするのではなく、多くの不要領域を含んでいると後の段落アライメントの際の効率・精度低下を招いてしまうので、HTML テキストの開始と終わりに非常に高い確率で存在する不要領域の塊を削除することを目的とする。

3.2　手法

明らかな不要領域は段落構成文字数が少ないものが多い。しかしながら、少数ではあるが文字数の多い不要段落、文字数の少ない文書段落も存在する。そこで、扱うテキストのモデルを、不要領域、文書領域共にかたまって存在し、不要領域はとりわけ文書の前後にかたまっていると仮定する（図 2）。実際のところ、不要領域は図 2 に示されている箇所以外にも存在するが、ここでは、基本的に、テキストの開始と終了部分の２つの塊のみに着目する。これにより、いわば問題を、不要段落・文書段落の個別的認定と、不要段落の塊と文書段落の塊との変化点抽出とを重ね合わせたかたちで捉えることになる。

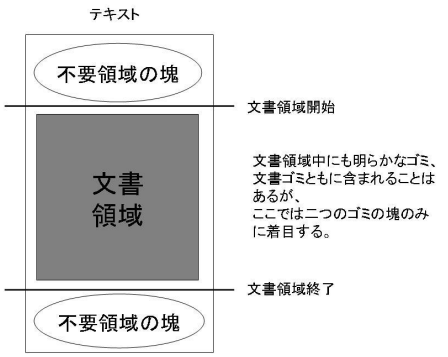

図 2. テキストのモデル

これにもとづいて、推定手法を極めて単純に、次のように定義する。すなわち、着目する段落と前後 r 段落ずつの文字数の和をとり、段落の文書（境界）度合 DD (DocumentDegree) を求め、その値を基準値 K と比較することで段落の性質を推定する。またこのとき、着目する段落そのものの性質を重点的に考慮するために、前後 r 段落の文字数に関しては、着目している段落との距離にもとづく重み付けを行う。

段落 n の DD の定義は次のようになる。

$$DD_n = \sum_{x=-r}^{r} (1 - |xw|) C_{n+x}$$

C_{n+x} は着目する段落 n から見て x 段落前（後）の段落の文字数、w は距離にもとづく重みコストである。現在は $r = 2$、$w = 0.2$ としている（図 3）。

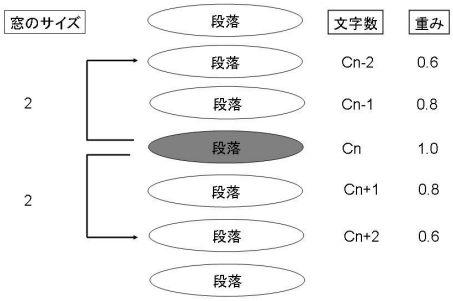

図 3. 重み付けの方法

基準値 K は、文字数 C を実際の文書段落サンプルの平均を参考に固定し[2]、次のように定義する。

$$K = \sum_{x=-r}^{r} (1 - |xw|) C$$

$DD_i \geq K$ ならば段落 i は文書段落、$DD < K$ ならば不要段落とする。

このようにして全ての段落についてその段落が不要段落であるか文書段落であるかを推定したあとで、テキスト全体の最初の文書段落を文書領域開始地点、最後の文書段落を文書領域終了地点とし、文書候補領域を抽出することで、テキストの開始と終わりに存在する不要領域の塊を削除する。この部分では、多少の不要領域は残っても、誤って文書領域を削除しないように配慮する。

4　段落アラインメント

文書領域候補抽出により残ったテキストに対して、すべての不要領域を取り除いてはいないことを前提に、DP マッチングを用いて段落のアラインメントを行う。このとき、日英の段落は一対一に対応していると仮定する。DP マッチングでは、文字数コスト、Number コスト、English コスト、Comment コストの 4 種類を用い、その総和の対応コストと削除コスト（挿入コスト）とを比較することで、コスト値の低い方を選択する[3]。

対応コスト

(1) 文字数コスト

日・英間で段落の持つ文字数の比が一定であると仮定し、文字数比 α を設定する。文字数比が α に近いものほどコストを低く与える。最大を 1 に設定する。

$$CharCost = \frac{|C_e - \alpha C_j|}{C_e}$$

[2]第 5 節で述べるが、実験と評価は $C = 40, 50, 60$ について行った。

[3]削除コストと挿入コストの関係は、各言語テキストにおける不要領域の存在確率に依存する。ここでは、英日間の対応付けを扱っているが、様々な言語対への拡張を想定して、英日固有の性質を考慮せずに、不要領域の存在確率はすべての言語で一定であると考え、削除コストと挿入コストに同じ値を用いる。

ただし C_j, C_e は日本語・英語の文字数を表す。

(2) Number コスト

段落内に数字が存在する場合、対訳関係にある段落内にも同じ数字が存在すると考えられる。

$$NumberCost = 0.2 \cdot \frac{Num_{je}}{max(Num_j, Num_e)}$$

ここで Num_j, Num_e は日英段落中に現れる数字の数、Num_{je} は両者で対応している数字の数である。

(3) English コスト

日本語の段落中に英単語（アルファベット列）が存在する場合、対訳関係にある英段落中に同じ英単語が存在すると考えられる。

$$EnglishCost = 0.2 \cdot \frac{Eng_{je}}{Eng_j}$$

ここで Eng_j は日本語段落中に現れるアルファベット列の数、Eng_{je} は日英で対応しているものの数である。

(4) Comment コスト

日本語の段落に角括弧（「」）などのコメント記号があるとき、対訳関係にある英語段落中には‘ ’や“ ”などの対応するコメント記号が存在すると考えられる。

$$CommentCost = \begin{cases} 0.0 & if \ \ Com_{je} \\ 0.2 & otherwise \end{cases}$$

Com_{je} はコメント記号が日英両段落に現れる場合である。

主な手がかりである文字数コストの最大値を 1 にし、他の三種類のコストの最大値を経験的な検討結果から 0.2 に設定してある。

削除コスト

対応コストに対して、閾値としての削除コストを設定する。実際には、これら 4 種の対応コストに関して、文書段落よりも不要段落の対応コスト値の方が低くなることも考えられる。このような問題を防ぐために、不要段落と文書段落はそれぞれある程度かたまって存在するという特徴を利用し、高削除コストと低削除コストという 2 つの削除コストの値を導入する。そして、DP マッチングの各ステップにおいて直前のステップが対訳関係にあるなら高削除コスト、削除対象となるなら低削除コストを用いる。

低削除コストは主に文書領域の前後の不要領域の塊、文書領域の開始段落の対応付けに影響を与えるコストである。つまり、明らかな不要段落の場合には、

対応コスト > 削除コスト ×2

文書領域の開始段落の場合には、

対応コスト < 削除コスト ×2

となるように設定する必要がある。

NumberCost、EnglishCost、CommentCost が全て最大値をとると仮定すると、

対応コスト = 文字数コスト +0.2×3

となり、低削除コストの条件は次のようになる。

不要段落同士の文字数コスト +0.6> 低削除コスト ×2> 文書領域開始段落の文字数コスト +0.6

また、20 件の URL 対で調べたところ、文書領域の開始段落の文字数コストは 0～0.25 であった。

以上より文書領域開始段落の対応付け漏れがないように、低削除コストを 0.3+0.25 に設定する。以下の実験では、高削除コストを変動させ、低削除コストとどれほどの差をもたせることで文書の固まりを精度よくアラインできるかを検証する。

5 実験と評価

実験には、10 の日本語側 Web サイトから取った対訳関係にある日英文書の URL 対 50 個を利用した。以下では、文書領域候補抽出と段落アラインメントそれぞれについて、実験結果を説明し考察を加える。

5.1 文書領域候補抽出

第 3 節で説明した手法で、基準値 K について、C = 40, 50, 60 の 3 つの値を用いて実験を行った。この値は、サンプル・データにもとづく日本語の文書段落の平均文字数を参考にして設定したものである。それぞれのパラメータ値で不要段落をどのくらい削除できたかを表 1 に示す。

表 1. 不要段落の削除率

C	60	50	40
日本語	83%	60%	46%
英語	77%	71%	45%

C = 60 のときに削除率は最も高いが、K の値が高くなりすぎ、わずかながら文書領域を不要領域と判断して削除してしまう場合がある。一方、C = 40 では、必要以上にゴミを文書領域と判断し、精度低下を招くことがわかった。C = 50 のとき、文書領域の抽出漏れがなく、高精度で三分の二程度の不要段落を削除することができている。単純な手法であるが、対象言語の文書段落の平均を参考に K を適切に設定することで、本手法がある程度有効であることが示された。とりわけ、不要領域の少ない URL に対しての効果はあまり大きくないが、大量の明らかな不要領域領域を持つ HTML テキストに関しては、段落アラインメントの効率・精度向上に十分貢献できるものである。

5.2 段落アラインメント

第 4 節で説明した高削除コストとして、0.55, 0.60, 0.65, 0.75, 0.85 の 5 つの値を用いて実験を行った。0.55 のときは高削除コストは低削除コストと同じである。また、本実験では一対一対応を前提としているので一対多対応の URL7 個（一つのサイト）を除いた 43 の URL を評価に用いた。一対多対応を含むサイト 7 以外の 9 サイトおよび全体について、それぞれのパラメータ値での対応正解率を表 2 に示す。

高削除コストが 0.55、つまり高削除コスト＝低削除コストの時正解率にもっとも低くなり、2 つの削除コストの差をもたせることで正解率が上昇しているのがわかる。中でも低削除コストに 0.2 の差を持たせた時に最も良い精度となる。コスト差 0.0 のときは、対訳段落よりも対応コストの低い不要段落と対応していたものが、高削除コストの値を上げることで、文書領域の「固まり」を持たせることができ、偶然の一致の不要段落

表 2. 段落アラインメントの精度

高削除コスト	0.55	0.60	0.65	0.75	0.85
サイト 1	91%	91%	100%	100%	100%
サイト 2	65%	71%	90%	90%	86%
サイト 3	40%	61%	76%	76%	62%
サイト 4	61%	61%	61%	86%	86%
サイト 5	11%	26%	28%	57%	50%
サイト 6	53%	57%	79%	80%	87%
サイト 8	84%	84%	89%	91%	91%
サイト 9	82%	82%	83%	83%	83%
サイト 10	71%	71%	76%	76%	64%
全サイト	56%	60%	68%	74%	71%

を削除し、対訳段落と対応付けすることができた。しかしコスト差 0.3 では精度が低下する場合が多く見られた。これは文書領域の「固まり具合」を高めすぎたために、無理矢理対応関係にない不要段落と対応付けしてしまったことによる。

各サイトにおいて精度の差が生じているのは、主な原因としてサブタイトル・会話部分で用いられる文字数の短い段落の対応付けに失敗しているためである。文字数が少なくなると日英の文字数比 $1:\alpha$ であるという法則性が崩れ、文字数コストの値が大きくなり、日、英ともに削除してしまう。また文書領域の「固まり」性を利用しているため文書領域中で頻繁に削除を行うと、その他の対応付けにも大きく影響を与えてしまう。文字数が少ないために対応付けに失敗した件数の占める割合を表 3 に示す。

表 3. 短い段落における失敗

高削除コスト	0.55	0.60	0.65	0.75	0.85
サイト 1	0%	0%	0%	0	0
サイト 2	4%	3%	3%	3%	3%
サイト 3	12%	12%	12%	12%	12%
サイト 4	3%	3%	3%	3%	3%
サイト 5	20%	20%	20%	20%	20%
サイト 6	18%	18%	18%	9%	9%
サイト 8	0%	0%	0%	0%	0%
サイト 9	13%	13%	13%	13%	13%
サイト 10	3%	3%	3%	3%	3%
全サイト	7%	7%	7%	6%	6%

失敗の他の原因としては、部分的な一対多対応、翻訳文内への訳者による注釈の書き込みによる文字数増加などがみられた。

これらの問題を解決する手段として、短い段落、注釈による文字数増加に関しては今後実装予定の辞書コストによる対応付けを、特に文字数が少ない場合は文字数コストの重みを下げるなどを、検討する必要がある。一対多対応に関しては DP マッチングのビーム幅を広げることで対応する予定である。ただし、ビーム幅を広げたときの各種コストの定義を検討する必要がある。

6 おわりに

本研究では、対訳関係にあるオンライン文書の URL を入力として、不要領域を削除し、段落単位で文書をアラインする基本的な手法を提案した。実際のデータにもとづく評価により、提案した単純な手法で、ある程度のパフォーマンスが得られることがわかった。

今のところ、ほかの言語対への拡張性を考えたときに、言語に依存しない処理でどの程度まで結果を出せるのか確認したかったこともあって、日英の言語情報を用いる処理は数値や有限の言語的特徴等、最低限しか使っていない。当然のことながら、対訳辞書を組み込めば、アラインメントの精度はさらにあがると考えられる。

また、不要な領域の削除は、原文書・対象文書ともに、それぞれが属するサイトにおける兄弟姉妹に相当する文書のタグ情報を活用すれば、さらに精度よく文書候補領域を抽出することができるかも知れない。ほとんどのサイトでは、どの記事にも同一のフォーマットと周辺情報が付与されているからである。

これらの情報を活用することは今後の課題としたい。また、最終的に我々がめざしているのは引用句や連語、固有名、専門語などのリサイクル可能な単位の抽出である。したがって、段落アラインメントを受けて、これらの単位を同定する必要があるが、一方で、こうした単位を同定するための手法自体が文書領域候補の認定や段落アラインメントに有用な情報を提供することも予測される。現在は基本的な手法を直列に組み合わせただけであるが、手法間の依存関係を考慮した全体の組み合わせについても、今後追求していきたいと考えている。

謝辞

本研究の一部は，日本学術振興会科学研究費補助金基盤 (A)「翻訳者を支援するオンライン多言語レファレンス・ツールの構築」（課題番号 17200018）の支援を得て行われた．

参考文献

[1] 影浦峡, 佐藤理史, 竹内孔一, 宇津呂武仁, 辻慶太, 小山照夫. 2006.「翻訳者支援のための言語レファレンス・ツール高度化方針」言語処理学会第 12 回年次大会発表論文集.

[2] 影浦峡, 関根聡. http://apple.cs.nyu.edu/akin/ に限定公開版がある。

[3] Veronis, J. (ed) 2000. *Parallel Text Processing*. Amsterdam: Kluwer.

[4] Melamed, D. 2001. *Empirical Methods for Exploiting Parallel Texts*. Cambridge, Mass: MIT Press.

[5] 北村美恵子, 松本祐治. 2006.「言語資源を活用した実用的な対訳表現抽出」 自然言語処理 13(1), p. 3–25.

[6] 内山将夫, 井佐原均. 2003.「日英新聞の記事および文を対応づけるための高信頼性尺度」 自然言語処理 10(4), p. 201–220.

[7] Resnik, P. and Smith, N. A. 2003. “The web as a parallel corpus,” *Computational Linguistics* 29(3), p. 349–380.