Web 上の兄弟ページを利用した対訳文書からの段落アラインメント

淺利 俊介[†]，竹内 孔一[†]，阿辺川 武[††]，影浦 峡[††]
[†] 岡山大学大学院自然科学研究科，[††] 東京大学大学院教育学研究科
[†]{syun1113, koichi}@cl.cs.okayama-u.ac.jp，[††]{abekawa, kyo}@p.u-tokyo.ac.jp

1　はじめに

現在オンラインの記事などの文書を翻訳し，自己サイトで公開するボランティア翻訳者の活動が活発である．そして翻訳者は，分野依存性の高い語や表現を翻訳する際に，その分野の既訳文書を参照するニーズを持っている．こうした背景から本研究では Web 上から既訳文書対を収集し段落アラインメントを行なうことで，専門用語や固有名詞，定型句の訳語候補を文脈まで考慮して提示する支援システムの構築を目指す．

Web 上の既訳文書からのアラインメントの先行研究として品川ら [1] があり，以前我々はそれを拡張した段落アラインメントを行なった [2]．しかし広告などの不要段落の除去が不十分であり，また辞書マッチの強化についても改善の余地があった．そこで本稿では新たな不要段落除去手法を加え，また辞書の補完を行なうことでも精度向上を図る．

Web 上の文書に対する不要部分の検出や部分情報抽出の先行研究には Web ページのレイアウトパターンに着目した韓ら [3] や教師あり学習を利用した中村ら [4]，ブラウザのレンダリング結果とヒューリスティクスに基づく不要部分削除を組み合わせた鶴田 [5] らがある．本稿ではボランティア翻訳者の Web ページでは複数の記事が翻訳され同サイト内で管理されている場合が多く翻訳元にも同様の特徴がある点に着目し，対象ページに対する兄弟ページを利用することで不要部分の除去を強化する．

2　システムの概要

本システムの大まかな流れを図 1 に示す．システムに与える入力は英・日 URL 対である．まずシステムは URL 対に対応する HTML テキストをダウンロードし，Parser を用いて文書領域を抽出する．ここで，本研究での対象となる Web 上の既訳文書は新聞記事等とは以下の点が大きく異なることに注意する．

- 体裁が様々であること
- 不要部分が多く存在すること

体裁の問題のうち半角全角や数値表現の揺れについては前処理を行い，カタカナの表記揺れについては 4.2.1 で示す手法で対応する．不要部分の存在は段落アライメントの精度低下に繋がるため，次の 3 つの手法を段階的に用いて除去を行なう．

1. 兄弟ページを利用した手法
2. Web 文書の特性と段落文字数を利用した手法
3. リンクに着目した手法

まず次節で示す兄弟ページを利用した手法により大まかな不要段落の削除を行なう．その後残った領域に対しては不要段落の多くが文章領域の先頭および後尾に塊となって存在しやすいという性質を利用し，不要段落の削除を行う．これは段落の文字数を利用することで対訳候補領域の開始と終了を決定する手法であるが，具体的な方法は品川ら [1] に譲る．そして残った対訳候補領域に対してリンクに着目した簡単な不要段落削除を行なう．Web ページに存在する広告は多くの場合リンクタグで囲まれている．この性質に着目し以下の特徴を持つ段落については段落対応付けから除外する．

- 全てがリンクで囲まれている段落
- リンク外が記号，数値表現のみの段落

以上の不要段落削除法を行なったのち最終的に残った対訳候補領域に対して 1 対 1 または 1 対多（多対 1）の段落アラインメントを行いシステムの出力とする．

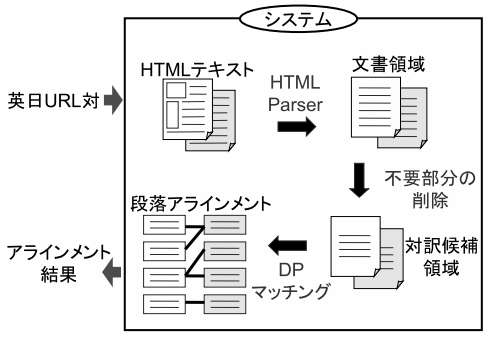

図 1: システムの流れ

3　兄弟ページを利用した不要段落削除

対訳関係にない不要段落を含めた段落アラインメントを行なうことは効率も悪く精度低下に繋がる．そこで段落アラインメントの前に，2 節で示した 3 つの手法による段階的な不要段落削除を行う．ここでは兄弟ページを利用した不要段落削除方法について述べる．

本研究で対象となる翻訳者の Web ページでは一記事が訳されているだけでなく，複数の記事が翻訳されサイトで管理されている場合が多い．そして管理された記事のページはそれぞれの体裁，形式がほぼ決まっている．また翻訳元である英語側のページにおいても，ニュースサイトなどの場合は同様に形式が決まっている．形式が決まっているものはサイトタイトル，ブックマーク，過去記事へのリンクなどでありこれらは同サイトの他ページと全く同一でありかつ対訳段落でない場合が多い．そこで段落アラインメントの対象となる Web ページから形式のよく似たページを発見し，その重なりを不要段落と認定する．

3.1　手法

まず対象ページのリンクを探索し，取得した兄弟ページを兄弟ページリストに追加する．そして得られたリストのそれぞれページと対象ページとの重なり

段落を検査し，重なっている段落数が最も多いページとの重なりを対象段落の不要段落と認定する．ここで兄弟ページとは対象ページと同サイトで同階層にあるページを指す．また対象ページと同サイトで一つ上の階層にあるページを親ページと呼ぶ．

その際 Web 上には膨大な量の兄弟ページへのリンクを含むページも存在することから処理時間の短縮のために兄弟ページリストにはある程度の数の制限を設けた[1]．兄弟ページへのリンクがリストの上限より多い場合「形式が似ているページは URL 名が似ている可能性が高い」という考えに基づき優先順位を決定する．URL 名の近さの指標には編集距離の考えを取り入れた．編集距離とはある語 A からある語 B に変換するために必要な「置換」,「挿入」,「削除」の編集回数を示したものである．リスト追加候補のそれぞれと対象ページの URL との比較を行い，編集距離が最も小さいものから順にリストに追加する．

また対象ページから直接兄弟にあたるページに辿れない場合，ひとつ上の階層である親ページ内のリンクから対象ページと兄弟にあたるページを再度検索する．直接兄弟ページを探す場合と同様，発見された場合は兄弟ページリストに追加していく．兄弟ページが多量に存在する場合には上記とは別の優先順位をつけてリストに追加する．ここでは「形式が似ているページは固まってリンク付けされやすい」という考えにより，親ページから対象ページへのリンク位置に近い兄弟ページから順位付けを行った．

4 段落アラインメント

対訳領域候補として残った英・日の段落に対し，DP マッチングを用いて段落アラインメントを行う．この時，英・日の段落対応関係を 1 対 1 および 1 対多（多対 1）対応を考える．DP マッチングに用いるコストとして対応コストと，不要な段落の削除を行う閾値の役割を果たす削除コストを用いる．削除コストの基本的な考え方は不要段落と対訳のある段落はそれぞれある程度固まって存在する特徴に基づいたものである．詳細は品川ら [1] に譲るとして，以下では各対応コストを説明した後，辞書を用いた対応方法および辞書の補完について述べる．

4.1 対応コスト

英・日の段落関係を示すコストとして，文字数比，数値表現，アルファベット表記，コメント，辞書マッチ，DP マッチングで 1 対多（多対 1）対応を行う際のビーム幅に関係するそれぞれのコストを規定する．このうち文字数比を表す Character コストの最大値を 1.0 とし，数値表現，アルファベット表記，コメントに関するコストを経験的な観点から 0.0〜0.2 の範囲とする．また Dictionary コストは 0 以下であり，対訳関係にあるほど小さな値を取る．

Character コスト

日・英間での段落の文字数の比は一定であると仮定し，文字数比が α に近いものほどコストを低く定める．

$$CharacterCost = \frac{C_e - \alpha C_j}{C_e}$$

C_e は英語段落の文字数を，C_j は日本語段落の文字数を表す．

Number コスト

英語，日本語それぞれの段落中に同じ数値表現が存在する場合，両段落が対応関係にある可能性が高くなるためコストを低く定める．

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

Num_{je} は英・日で対応している数値表現の数，Num_j，Num_e は日・英それぞれの数値表現の数を表す．

English コスト

日本語段落中にアルファベット表記が存在し，英語段落に同じ英単語が存在する場合，両段落が対応関係にある可能性が高くなるためコストを低く定める．この English コスト値の式は以下となる．

$$EnglishCost = 0.2(1 - \frac{Eng_{je}}{Eng_j})$$

ここで Eng_{je} は英・日両段落に存在する英単語の数，Eng_j は日本語段落に含まれる英単語の数を表す．

Comment コスト

「,」や”，”のようなコメント部分が英日段落それぞれに存在する場合，両段落が対応関係にある可能性が高くなるためコストを低く定める．

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

Com_{je} は英・日両段落にコメントが存在する場合を示す．

Dictionary コスト

ストップワードを除く英語段落の自立語で辞書引きを行い，対訳候補が日本語段落の語と一致する場合は対応コストを下げる．

$$DictionaryCost = -\beta\frac{Dic_{je}}{Dic_e}(TFIPF_j + TFIPF_e)$$

β は Dictionary コストとその他コストの重みを決定するものである．前回 [2] の結果から実験では $\beta = 0.3$ として段落アラインメントを行った．また Dic_e は実際辞書引きを行った英段落の自立語数，Dic_{je} は，辞書引きの結果日本語段落の語と一致した数である．$TFIPF_j$，$TFIPF_e$ は対訳関係にある日・英の語の TFIPF 値[2]を表す．辞書には英辞郎（英語見出し：約 20 万エントリ）を使用した．

Beam コスト

1 対多（多対 1）対応の段落アラインメントを行う際に DP マッチングのビーム幅を拡張するが，このときに 1 対多（多対 1）対応は起こりにくいという観点から，Beam コストを設定する．

$$BeamCost = \gamma(BeamWidth - 1)$$

BeamWidth は DP マッチングの際のビーム幅を表す．今回は γ の値を 0.2 とし実験を行った．

[1]本稿の実験では 12URL とした．

[2]TFIDF を文書単位でなく段落単位に適用したものを TFIPF とした．

4.2 辞書を利用した対応法と辞書補完

前節の辞書コストを計算するために既訳文書の候補段落対に対して英辞郎を利用して，単語の対応を求める．しかしながら日英の品詞が異なる場合がある，固有名詞，新語などは対応不可能であることなどから出来る限りの辞書補完を行なう．以下では，日英の単語対応方法および辞書補完方法について具体的に述べる．

まず英語段落中の語について TreeTagger [3]を用いて解析を行い，自立語の場合のみ原形で辞書引きを行う．日本語段落の語についても Sen[4]にて形態素解析を行い，英辞郎による訳語候補が日本語段落の語の原形と一致する場合を辞書マッチとする．なお日本語段落側は複数形態素との一致を許している．

ここで英日の品詞不一致への簡単な補完を行なう．例えば「infiltrate」という語には「浸透する」,「侵入する」という動詞表現しか対訳候補存在しないが，日本語段落中に出現する「浸透 工作」には対応付けが出来ない．そこで「サ変名詞＋する」の対訳候補が存在する場合は「する」を除去したものも対訳候補に追加する.「safe」の対訳候補である「安全な」,「無事に」などについても同様に，名詞化した「安全」,「無事」を対訳候補に加える．

4.2.1 カタカナに着目した辞書補完

より多くの辞書対応が可能となるようにカタカナに着目した英辞郎の訳語候補に対する補完を行う．本研究では英語を翻訳した日本語ページを対象としているため，多くのカタカナが出現する．しかしカタカナには発音，翻訳者の癖などから発音的には誤りである事例も含め多くの表記ゆれが存在する．そのため単純に辞書マッチを行うことはできない．

また製品名や企業名などの固有名詞，新しい語については辞書にも登録されていない．そしてこれらの語は日本語側ではカタカナで記述される場合が多い．そこで辞書で対応し切れなかったこれらの語とカタカナ語との対応を目指す．

ここでは辞書の訳語候補のカタカナと日本語段落のカタカナの間で起こる表記揺れへの対策と，固有名詞や新語のカタカナへの対応付けについて述べる．

カタカナの表記揺れへの対応

英語辞書の候補にカタカナを含む場合，そのカタカナから起こりうる様々な表記揺れのパターンを生成することで日本語段落との対応を図る．例として「ヴェトナム」から生成した表記揺れパターンを以下に示す．

- ヴェトナム，ベトナム，ヴェトナムー，ベトナムー

この時「ヴェトナムー」など誤りである表記揺れのパターンも生成されるが，その多くは言語として存在しない語であり対象となる Web 文書にそのパターンが出現する可能性は少ないと考えられる．

この結果，例えば辞書の対訳候補中に「ヴェトナム」，本文中に「ベトナム」が出現した場合でも辞書マッチが行なえる．

カタカナと Alphabet の対応付け

本研究では辞書として英辞郎を使用するが，企業名や製品名をはじめとする固有名詞については登録されてない．また別に辞書を用いるとしても新しい語については対応しきれない．そしてこのような固有名詞，新しい語はカタカナで示されることも多いため，これについての対応付けを考える．

この処理は辞書の補完が目的であり，また固有名詞や新語を対象とするため，英語段落では「辞書に登録がなく」かつ TreeTagger による解析で原形が「大文字から始まる語」を抽出する．日本語側ではカタカナで構成される単語を対象とする．また英日とも辞書対応された語に対してはこの処理は行なわない．

まず日本語段落中に含まれるカタカナをローマ字表記への変換を行なう．そして変換後のローマ字アルファベット列から英語とローマ字の関係上起こりうる展開を行なう．

最後に編集距離を用いて，展開されたそれぞれのアルファベット列と英語段落の対象語との距離を測る．こうして得られた最短の編集距離を元のカタカナと英語段落の対象語との距離とする．そしてその距離に対し閾値をもうけ，その閾値以下であるなら対応付けを行なう．今回は辞書補完が目的であるためスコアとしては Dictionary コストに含み，単語の重みも 4.1 で定義した TFIPF を使用する．日本語側「ブログ」，英語側「blog」との対応過程における編集距離を表 1 に示す．この例では途中で編集距離が 0 となるため処理を終え，最終的な語間の距離は 0 となる．

今回は語間の距離の閾値を 1 とし，またゴミを減らすため日本語文字が 2 以上の場合のみ対応という制限を加えた．

表 1: 変換の過程　例：「ブログ」と「blog」

アルファベット列	「blog」との編集距離
burogu	3
bulogu	2
brogu	2
:	:
brog	1
blog	0

5 実験および考察

段落アラインメントの精度を調べるために，実際の Web 上の既訳文書対に対し実験を行う．3 節で示した兄弟ページによる不要段落の除去を行なうことによる結果への影響を調べるとともに，辞書マッチを行なう際の簡単な位置情報の導入，辞書マッチの割合による対応付けのフィルタの追加を行い精度向上を図る．

5.1 実験

対訳関係にある日・英の URL28 対（7 つの Web サイトから 4 ページずつ）に対し段落アラインメントを行った．兄弟ページによる不要段落の除去が結果に与える影響を調べるため，結果の比較を行なう．表 2 に 28URL 対の平均の結果を示す．

表 2: 段落アラインメント結果

兄弟ページ手法	recall	precision
使用	0.7351	0.6774
不使用	0.7342	0.6099

[3]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

[4]https://sen.dev.java.net/.

また辞書マッチの際に語の位置情報による簡単なフィルタをかけ，それが結果にどのような影響を与えるのかを調べる．これは 例えば英語段落中の 1 文目に含まれる語と日本語段落中の 15 文目に含まれる語との辞書マッチなど，距離的にあり得ない対応付けを防ぐものである．このフィルタによってより正しい 1 対多 (多対 1) の対応付けを行なえるという効果が期待できる．ここでは英，日それぞれの平均の文数，1 対多対応の割合などから 10 文以内の英日間の辞書対応のみ許すこととする．兄弟ページ手法を使用したものに今回の位置情報を加えた結果は recall は 0.7377，precision は 0.6801 となった．

またこれにアラインメントの際に辞書マッチの割合によるフィルタを加えた実験を行なう．これは正しく対応付けされた段落対は一定割合以上の辞書マッチが起こるという考えに基づくものである．いくつかのサンプルページの正しい対訳対を調べたところ，最も辞書マッチの割合が低いもので 0.3 程度であった．そこで辞書マッチの割合が 0.1 以下の場合対応付けを行なわないという緩い制約を設けた．その結果 recall が 0.6921，precision が 0.7223 となった．

5.2 考察

表 2 から兄弟ページによる不要段落の除去が特に precision の向上に効果があることがわかる．個別の結果をみてみると兄弟ページを使用した方が recall が下がっている URL 対もあるが，これは重なりにより除去された部分にタイトル，見出しなどの対訳段落を含む場合があるためである．しかしこれらは語数の少なさ，それに伴う文脈情報の少なさから翻訳者が求める「文脈を含めた既訳情報」とは異なる．そしてタイトル，見出しで出現した語は本文中でも出現しやすいと考えられるため，重複を考えると対応としては不要といえる．ここで対象ページの内，重なりを持つ兄弟ページがどれほどの割合で取得できているかを調べたところ，56(28 × 2)URL のうち 40URL においてなんらかの兄弟ページが取得でき，そのうち 33URL で対象ページとの重なりがあり不要段落を削除できた．不要段落が削除できなかったものの原因はリンクが貼られていない，リンク切れ，形式が異なり重なりが無いなどの理由であった．また 56URL の平均をみると全段落中の約 59 %が広告などの不要段落であった．2 節で示したように不要段落削除法を組み合わせることで，この不要段落のうち 81 %を除去し DP マッチングを行なうことが可能となった．

また位置情報を取り入れた辞書マッチでも若干の精度向上が見られた．今回は 10 文という数値を使用したが，この最適値を探すことができれば更なる向上が可能と考えられる．

辞書マッチ割合によるフィルタでは precision が大きく向上した．recall は低下したが，Web 上から多くの既訳情報を収集できることを考えると recall よりも precision を重視すべきといえるため，このフィルタは有益であるといえる．個別 URL 対の結果を確認すると半分にあたる 14URL で recall は変化せず，大きく recall が低下した URL 対の数は 0.2 下がったものが 1URL，0.15 下がったものが 2URL と少なかった．そのためこのフィルタを導入しても recall の極端な低下によるデータ収集効率の悪化は起こりにくいと考えられる．今回は辞書マッチの割合のみに着目したが，今後は英日の段落文字数比によるフィルタを加えることも検討中である．

6 まとめ

本研究は Web 上の既訳文書対の段落アラインメントを行なうことで，ボランティア翻訳者に対し文脈情報を含めた未知の専門用語，固有名詞，定型句に対する訳語候補の提示を目指すものである．DP マッチングによる段落アラインメントの精度向上を図るため，前処理として兄弟ページを利用した不要段落の削除を行なったのち，言語に関する様々な特徴を利用しコストを定め対応を行うシステムを提案した．実際の既訳文書 28 対を対象に段落アラインメントを行ったところ，兄弟ページによる不要段落削除を行なわない場合 recall は 0.7342，precision は 0.6099 であったが，行なった場合は recall が 0.7351，precision が 0.6774 と大きな向上が見られた．そして位置情報を取り入れた辞書マッチや辞書マッチによる対応付けフィルタでも有益な結果が得られた．これについては段落文字数比を考慮したフィルタを組み合わせることなどで改善が期待できる．またどのように見やすい形で翻訳者に提示するのかというインタフェースについても検討したい．

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）の支援により行われた．

参考文献

[1] 品川哲也，森辰則，影浦峡，“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”，言語処理学会第 12 回年次大会発表論文集，pp.520-523，2006.

[2] 淺利俊介，竹内孔一，阿辺川武，影浦峡，“Web 上の既訳文書を対象とした段落アラインメント”，言語処理学会第 14 回年次大会発表論文集，pp.337-340，2008.

[3] 韓浩，徳田雄洋，“Web 部分情報抽出システムとその応用”，日本ソフトウェア科学会第 23 回大会論文集，2006.

[4] 中村達也，白井清昭，“ウェブページにおける非コンテンツ領域の検出”，言語処理学会第 13 回年次大会発表論文集，pp.234-237，2007.

[5] 鶴田雅信，増山繁，“未知のサイトに含まれる Web ページからの主要部分抽出手法”，言語処理学会第 14 回年次大会発表論文集，pp.197-200，2008.

[6] 内山将夫，井佐原均，“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），pp.201-220，2003.

[7] 宇津呂武仁，池田浩，山根正也，松本裕治，長尾眞，“対訳辞書を用いた対訳文対応および未知訳語の推定”，「自然言語処理における実動」シンポジウム論文集，pp.140-143，1993.