対訳複単語表現を利用した法令文の統計的機械翻訳

坂本 聡美[1]　小川 泰弘[1,2]　大野 誠寛[1,2]　中村 誠[3]　外山 勝彦[1,2]
[1] 名古屋大学 大学院情報科学研究科　[2] 同 情報基盤センター　[3] 同 大学院法学研究科
{satomi,yasuhiro}@kl.i.is.nagoya-u.ac.jp

1　はじめに

現在、日本の法情報の国際的発信が進められている。主要な法令は既に英訳され、法務省の日本法令外国語訳データベースシステム (JLT)[1] で公開されている。しかし、JLT で英訳が公開されている法令の数は十分ではなく、法令の公布から英訳の公開までにかかる時間も長いという問題がある。これらの問題に対応するため、統計的機械翻訳 (SMT) による法令翻訳の支援を考える。

そもそも、法令文は一般的に読み難く、一読して理解することが難しい。その原因として、法令文の一文の長さ、法令文に出現する専門用語、複雑な表現、複雑な依存構造などがあげられる。これらの要素は互いに関係がある。例えば、法令文では文の意味を正確に表現するため、一つの語に多数の修飾語が付加された長い表現が用いられる。これにより一文は長くなる傾向にある。また、法令文で用いられる接続詞は、その出現順序や意味が厳密に決められた法令の専門用語である。一見複雑に見える依存構造も、専門用語の知識があれば理解しやすいが、こうした知識を得るのは容易ではない。また、複雑な構造を持つ文もまた一文が長くなる傾向にある。

このような法令文の難しさは SMT の精度にも影響する。SMT では自動単語アライメントによって表現の翻訳規則を学習する。アライメントは文の長さや表現の複雑さに影響されて精度が低下するため、法令文の難しさにより正しい翻訳規則の学習が困難になり、結果として翻訳精度は低下する。

本稿では、法令文の難しさを引き起こす原因のうち、法令の専門用語や複雑な表現に着目する。このような表現は複単語表現 (MWE) として捉えられる。MWE とは「単語の境界を横断する意味をもつ慣用的な表現」[2] であり、一般には慣用句や動詞句、名詞句、前置詞句などを指す。本稿では MWE を広い意味で捉え、慣用的な表現に限らず、機能表現や特定の意味を持たない単語列も MWE であるとする。

MWE の中には構成的ではない方法で翻訳されるものがある。例えば、「民事訴訟法」を形態素解析すると、「民事」「訴訟」「法」の 3 単語に分割される。しかし、「民事訴訟法」の対訳 “Code of Civil Procedure” には、その構成語である「訴訟」の対訳 “litigation” が含まれておらず、対訳は構成的でない。構成的に翻訳されなかった表現は自動単語アライメントのノイズとなりやすく、翻訳精度低下の原因となる。法令文には専門用語や定形表現といった形で多くの MWE が出現するため、これを適切に扱うことにより翻訳精度の向上が期待できる。

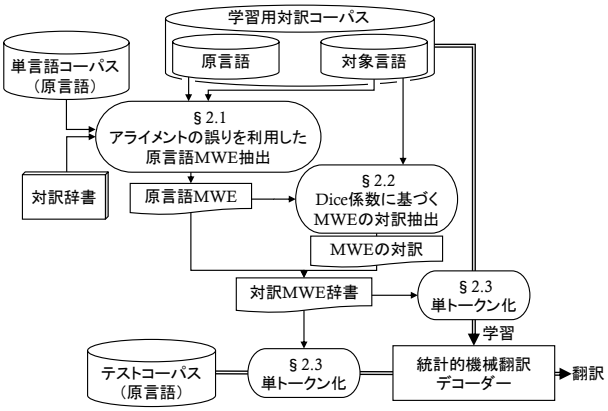

図 1: 提案手法の概要

本稿では、法令文に出現する MWE を考慮して SMT の精度を改善する枠組みを提案する。提案手法では、GIZA++のアライメントが誤った箇所から MWE を抽出する Tsvetkov らの手法 [3] を用いて原言語の MWE を抽出する。MWE の対訳は、GIZA++のアライメントによらず、Dice 係数に基づいて抽出する。作成した対訳 MWE 辞書は、Pal ら [1] の提案する MWE の単トークン化によって SMT の学習・デコードに用いる。

提案手法を用いた SMT は、ベースラインと比較して有意に翻訳精度を改善した。さらに、この対訳 MWE 辞書の特徴を考慮し、学習データへの単トークン化の適用方法について検討した。

2　提案手法

提案手法の処理手順を図 1 に示す。まず、アライメントの誤りを利用した Tsvetkov らの手法 [3] を用いて原言語の MWE を抽出する (2.1 節)。次に、原言語 MWE の対訳を Dice 係数に基づいて抽出する (2.2 節)。最後に、作成した対訳 MWE 辞書を用いて学習データの MWE を単トークン化 [1] し (2.3 節)、SMT に用いる。以下では、各処理について詳述する。

2.1　アライメントの誤りを利用した原言語 MWE 抽出

原言語 MWE の抽出には Tsvetkov ら [3] の手法を使用する。この手法は、対訳テキストにおいて、1 単

[1] http://www.japaneselawtranslation.go.jp/

語対 1 単語のアライメントがされない表現は、すべて MWE の候補であるという考えに基づいている。そこで、アライメントを辞書で確認し、正しい 1 単語対 1 単語の対応である箇所を全て取り除き、残りの単語列から MWE を抽出する。単語列中のどの部分が抽出すべき MWE であるかを判別するため、単語列の任意のバイグラムに対して自己相互情報量 PMI^k を式 (1) により計算する。

$$\mathrm{PMI}^k(x,y) = \frac{P(x,y)^k}{P(x)P(y)} \tag{1}$$

ここで、$P(x), P(y)$ はそれぞれコーパス中のユニグラム x, y の出現回数、$P(x,y)$ はバイグラム xy の出現回数、k は重みである。PMI^k が閾値以上の場合は連接する表現として認め、閾値を下回る場合は MWE の切れ目であるとする。最後に、2 単語以上の単語列を MWE として抽出する。

2.2 Dice 係数に基づく MWE の対訳抽出

Tsvetkov らの手法では、単言語 MWE の対訳を抽出する際、GIZA++によって付与されたアライメントを辿って対訳とする。しかし、元々この手法は GIZA++ が誤った箇所を対象に MWE を抽出しているため、このような方法で抽出された対訳の精度は信頼できない。実際、4 単語より長い対訳はノイズが多いと Tsvetkov らも指摘している。さらに、ある MWE が異なる対訳文でそれぞれ別の表現に翻訳されていた場合、それらに対するランク付けの方法を Tsvetkov らは示していない。本研究が対象とする法令文には長い表現が出現するため、対訳を 4 単語までに制限することは望ましくない。また、同じ原言語 MWE でも、出現する法令の分野が異なれば異なる表現へ翻訳されている可能性があり、表現の適切さを示す指標が必要である。

そこで、Tsvetkov らの使用した対訳抽出手法に代えて、Dice 係数に基いて対訳を抽出する。Dice 係数とは二つの表現の間の類似度の指標であり、式 (2) によって計算する。

$$Dice(x,y) = \frac{2 \cdot freq(x,y)}{freq(x) + freq(y)} \tag{2}$$

ここで、$freq(x)$ と $freq(y)$ はそれぞれ原言語文中での表現 x の出現数と対象言語中での表現 y の出現数である。$freq(x,y)$ は対訳文に x と y が同時に出現する回数である。提案手法では、x に抽出した原言語 MWE を与え、x と Dice 係数類似度が最も高くなる 1 単語以上の表現 y を対訳コーパスから探索する。このとき、y の Dice 係数の値が閾値以上ならば対訳として抽出する。抽出された対訳を含まない文を集めて類似度最高の y を探索し、これを繰り返す。これにより、ある原言語 MWE に対して複数個の対訳を抽出することが可能である。

2.3 対訳 MWE の単トークン化

抽出した対訳 MWE を SMT に対して用いるため、Pal ら [1] の提案した単トークン化の手法を用いる。単トークン化は MWE の単語間の空白記号をアンダースコア “_” で置き換える処理である。これにより文中で空白区切りの複数の単語からなる表現として存在していた MWE は、アンダースコアを含む新しい単語に変わる。MWE をその構成単語とは無関係の一単語としてシステムに認識させることになり、MWE を MWE として学習・デコードすることが期待できる。さらに、一文あたりの単語数が減ることで自動単語アライメントの探索空間を削減できる。その結果、MWE 以外の部分についてもアライメントの質の向上や翻訳精度の向上が期待できる。

対訳 MWE 辞書を用いた単トークン化を学習データの前処理として行う。Pal らの手法に則り、学習データにおいて対訳 MWE が原言語側と対象言語側に同時に出現する対訳文のみ単トークン化の対象とする。なお、文中における対訳 MWE の出現箇所の検索は、対訳の Dice 係数の値によらないで最長一致により行う。

3 MWE 辞書を用いた翻訳実験

本節では、提案手法の有効性を検証するため、翻訳実験を行う。学習データから MWE を抽出して対訳 MWE 辞書を作成し、単トークン化を行ったモデルをベースラインと比較する。

3.1 MWE 辞書の作成

対訳 MWE の抽出元となる対訳コーパスには JLT に掲載されている法令の対訳文 (以下、JLT コーパスと呼ぶ) から 151,951 文を用いる。これは翻訳実験のテストデータとして JLT コーパスからランダムに選出した 15,026 文を除いた残りである。アライメント結果から 1 単語対 1 単語の対訳を確認するために使用する対訳辞書には、英辞郎に人手で作成した漢数字・ローマ数字対訳を追加したものを用いる。なお、対訳コーパス中に出現した対訳辞書の見出し語数は 5,618 単語であった。

対訳コーパスと対訳辞書には小文字化、トークン化、レンマ化の前処理を実行する。小文字化とトークン化では、日本語部分に対しては MeCab を、英語部分に対しては Moses 付属のスクリプトを用いる。レンマ化には、日本語部分に対しては MeCab を、英語部分に対しては Ruby Lemmatizer を用いる。GIZA++は長すぎる文を扱えないため、対訳コーパスから片方の文が 80 単語以上となる対訳文を削除する。

提案手法により法令対訳コーパスから対訳 MWE を抽出する。予備実験の結果をもとに PMI^k の重み k は 2.7、閾値は 1 に設定する。なお、句読点混じりの MWE にはノイズが多かったため、このような候補を除いてから各 MWE の英語対訳を Dice 係数を用いて抽出する。

表 1: 翻訳の BLEU スコア

	0.1	0.3	0.5	0.7	0.9
Dice モデル	**31.19**	30.75	**30.98**	**31.26**	30.51
Manual モデル	30.89	**30.97**	**31.30**	**31.35**	30.70
ベースライン	30.32				

句読点混じりの MWE を除いた後の日本語 MWE は 2,829 個であった。このうち 544 個の日本語 MWE は、Dice 係数の閾値を 0.1 に設定しても対訳を得ることができなかった。

3.2 MWE 辞書の SMT への適用

作成した対訳 MWE 辞書を SMT に用いる。Dice 係数に基づいて抽出された英語対訳は必ずしも正しい翻訳ではないことを鑑み、人手で対訳を修正したモデルも用意して比較対象とした。

学習データには 3.1 節で MWE の抽出元に用いたものと同じ対訳コーパス 151,951 文を使用する。この学習データから (1) ベースライン、(2)Dice、(3)Manual の 3 種類のモデルを作成する。学習データに小文字化とトークン化をした後、ベースラインはこれをそのまま用いてモデルを作成し、Dice モデルと Manual モデルはさらに対訳 MWE を用いて単トークン化の処理をしてからモデルを作成する。Dice モデルは、Dice 係数の閾値を 0.1 から 0.9 まで変化させて 5 種類のモデルを用意する。Manual モデルは、Dice モデルの各閾値で得られた MWE の対訳に人手で修正を加えることで作成する。Manual モデルも Dice モデルと同様に閾値の異なる 5 種類のモデルを用意する。モデル作成時には、GIZA++が長すぎる文を扱えないため、片方の文が 80 語以上となる対訳文は削除する。

テストには 15,026 文の日本法令文を用いる。これは JLT コーパスから学習データを抽出する際にあらかじめ除いておいたものである。テストデータも学習データと同様に小文字化、トークン化の処理を施す。Dice モデルと Manual モデルのテスト時には MWE の単トークン化も施す。

翻訳モデルの作成には GIZA++を、言語モデルの作成には SRILM を、翻訳のデコーダーには Moses をそれぞれ用いた。翻訳結果の自動評価には BLEU を用いた。翻訳結果の中に “_” が付いて単トークン化された MWE がある場合は、これを半角スペースに戻して評価した。

3.3 結果と考察

表 1 に各モデルの自動評価の結果を示す。このうちベースラインより有意に ($p < 0.05$) 評価が向上した結果は太字で表記した。

Dice モデルと Manual モデルは、閾値 0.7 を頂点として、いずれもベースラインより有意に BLEU スコアを改善した。Dice 係数の閾値を下げると原言語 MWE に対して抽出される対訳の数は増える。しかし、このような対訳数の増加はスコアの改善に反映されなかった。Dice モデルにおいては、低い閾値で獲得された対訳の精度の低さにより、翻訳器の学習とデコードに支障を来たしたものと考えられる。しかし、人手による対訳付けにより対訳の精度が担保されている Manual モデルにおいても、閾値を下げてより多くの対訳 MWE を追加した場合に翻訳精度は改善していない。この理由として、対訳のバリエーションが影響していると考える。ある MWE に対して時制や動詞の態、単複の形によって複数の対訳が存在するとき、Dice 係数のスコアは低くなりがちである。このような対訳のバリエーションは、形が異なっていても全て正しい対訳といえる。しかし、人手で対訳を修正してもこれを網羅することは容易ではない。単トークン化は、学習データにおいて対訳 MWE が原言語側と対象言語側に同時に出現する対訳文に対してのみ実施される。原言語 MWE に対して対訳を網羅できていない対訳 MWE 辞書を用いる場合、単トークン化される場所とされない場所が発生し、探索空間の削減に悪影響を及ぼす可能性がある。したがって、Dice 係数の閾値が低いということは、Manual モデルにおいてもスコアの改善につながらなかったと考えらえる。結果として、本手法において対訳 MWE を活用するためには、Dice 係数の閾値を 0.7 に設定するのが適切である。

先に述べたように、Dice 係数によって抽出された対訳は必ずしも正解ではないが、閾値 0.7 では Dice モデルと Manual モデルの評価結果は同等である。そこで、閾値 0.7 で抽出された対訳を調べたところ、完全な正解ではないが部分正解であるものが存在した。例えば日本語 MWE「産前 産後」に対して JLT コーパス中で使用されている正しい対訳は “before or after childbirth” または “before and after childbirth” である。一方、Dice 係数により得られた対訳は “after childbirth” であり、正しい対訳の共通の部分文字列を抽出していた。このように Dice 係数により抽出された対訳は正しいとは言えないが、正しい対訳の部分文字列であるため、対訳文内で日本語 MWE と共起する可能性が高い。部分正解の MWE でも、単トークン化することにより、単語アライメントにおける探索空間の削減とアライメント誤りの低減に貢献できる。その結果、SMT の精度向上につながったものと考えられる。よって提案手法は、人手による対訳の修正無しでも修正有りの場合と遜色ない性能をもつと言える。

4 提案手法の改良

提案手法を改良するため、学習データの単トークン化の方法を検討する。対訳 MWE が原言語側と対象言語側に同時に出現する対訳文のみ単トークン化の対象にする Pal らの手法に対し、対訳文における出現状況を確認しないで単トークン化する素朴な方法を提案し、その有効性を実験によって確認する。

4.1 学習データの素朴な単トークン化

Pal らは、学習データにおいて、対訳 MWE が原言語側と対象言語側に同時に出現する対訳文のみ単トークン化の対象にした [1]。条件に合わない対訳文を対象から外すことにより、過剰に単トークン化することを防げる。このように、対訳文の原言語側・対象言語側の両方に対訳 MWE が含まれている場合にのみ単トークン化する方法を、選択的な単トークン化と呼ぶ。

しかし、提案手法では Dice 係数により対訳を抽出している。Dice 係数により得られる対訳は必ずしも正確ではないが、共起度の比較的高い単語列である。一方、真に正しい対訳でも出現数が少ないとその Dice 係数は低くなり、対訳として抽出されない。このように、作成した対訳 MWE 辞書は対訳を網羅していない。この辞書を基準にして選択的に単トークン化すると、本来単トークン化するべき MWE を含む文が単トークン化の対象から外れ、MWE の学習を不当に妨げる可能性がある。

さらに、テストデータには参照すべき対訳文が存在しない。Pal らはテストデータの単トークン化方法について言及していなかった。そのため、テストデータに対しては、提案手法では学習データで単トークン化された全ての MWE を単トークン化するという方法を採用した。しかし、これには問題がある。学習データでは対訳文によって同じ表現でも単トークン化される箇所とされない箇所が存在し、単トークン化後はそれぞれ別の表現として学習される。一方、学習データで一度でも単トークン化された MWE はテストデータにおいて全て単トークン化された状態で翻訳されるため、学習データで単トークン化されなかった箇所の学習結果はテストデータの翻訳に影響しない。学習データが少なかったり、対訳 MWE 辞書が不十分で MWE が出現する文の一部しか単トークン化の対象にならなかった場合、MWE を十分に学習できない恐れがある。

本節では、学習データに対して素朴な単トークン化をすることを試みる。素朴な単トークン化とは、対訳 MWE 辞書に登録されている全ての表現を対訳文に同時に出現しているかどうかに関わらず単トークン化することを意味する。これにより辞書が網羅できていない部分をカバーし、学習データ中で単トークン化済み MWE の学習可能な箇所が増えると期待される。

4.2 実験設定

3 節の実験と同じ対訳コーパスを用いて実験する。比較のため JE モデル、JA モデル、EN モデルの 3 種類を用意する。JE モデルでは日英対訳 MWE を用いて学習データを素朴に単トークン化する。JA モデルは日本語 MWE だけを用いて、EN モデルは日本語 MWE の英語対訳だけを用いて学習データを素朴に単トークン化する。全てのモデルに対して Dice 係数の閾値を 0.1 から 0.9 までの 0.2 刻みで変化させた 5 種類を用意し実験した。

表 2: 素朴な単トークン化による翻訳の BLEU スコア

	0.1	0.3	0.5	0.7	0.9
JE モデル	31.33	**31.37**	31.47	31.26	30.48
JA モデル	30.80	30.79	31.09	30.95	30.19
EN モデル	30.20	30.08	30.52	30.58	30.42
Dice モデル	31.19	30.75	30.98	31.26	30.51

4.3 結果と考察

各モデルの翻訳結果を BLEU で評価した結果を表 2 に示す。参考のため 3.3 節の Dice モデルの結果を併記した。各閾値において Dice モデルより有意に ($p < 0.05$) 評価が向上した結果は太字で表記した。JA モデルと EN モデルはいずれも Dice モデルの翻訳精度に及ばなかった。両モデルの使用した MWE 辞書は単言語であり、Dice モデルの対訳 MWE 辞書の片側しか使用していないことが原因であると考えられる。

対訳 MWE 辞書を用いて素朴に単トークン化した JE モデルは、閾値 0.3 でのみ Dice モデルより有意に翻訳精度を改善した。比較的高い閾値である 0.7、0.9 では学習データの素朴な単トークン化の効果は見られなかった。Dice 係数が十分高い対訳は学習データの対訳文に共起する可能性が高い。そのため、選択的な方法と素朴な方法の違いによる単トークン化の結果に差が生じなかったと考えられる。

5 おわりに

本稿では、法令文に出現する MWE を考慮して SMT の精度を改善する枠組みを提案し、ベースラインと比較して有意に翻訳精度を改善可能であることを示した。さらに、提案手法により作成される対訳 MWE 辞書の特徴を考慮し、学習データへの単トークン化の適用方法について検討した。共起度が高くない表現を含む対訳辞書を使用する際は、対訳文へ同時に出現するかどうかを考慮せず、素朴に単トークン化する方が精度をより改善する可能性があることを示した。今後は、対訳 MWE 辞書のより適切な構築方法を検討し、どのような MWE が翻訳精度向上に寄与するかを評価する予定である。また、今回は MWE が多く出現すると考えられる法令文に対して手法を提案したが、他のドメインに対する提案手法の有効性も調べる予定である。

参考文献

[1] Santanu Pal, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. MWE alignment in phrase based statistical machine translation. In *Proc. MT Summit XIV*, pp. 61–68, 2013.

[2] Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proc. CICLing2002*, pp. 1–15, 2002.

[3] Yulia Tsvetkov and Shuly Wintner. Extraction of multi-word expressions from small parallel corpora. In *Proc. Coling 2010: Posters*, pp. 1256–1264, 2010.