対訳文書からのモデル学習：日韓・韓日統計的機械翻訳

丁 塵辰 †　内山 将夫 ‡　吉田 光男 †　山本 幹雄 +
†+ 筑波大学 システム情報工学研究科　‡ 情報通信研究機構

†{tei,ceekz}@mibel.cs.tsukuba.ac.jp　+myama@cs.tsukuba.ac.jp
‡mutiyama@nict.co.jp

1　はじめに

日本語と朝鮮語（韓国語）は、語彙及び構文の類似度が高い言語である。スペイン語とカタルーニャ語も語彙及び構文の類似度が高い言語であり、これらの言語間では文字単位による翻訳が可能であると示唆されている [1]。同様に、語彙及び構文の類似度が高い言語である日韓・韓日間においても文字による翻訳が可能であると考えられる。また、一般的な統計的機械翻訳システムは文単位の対訳コーパスを必要とする。対訳文対コーパスの収集には、一般的に生の対訳データから辞書を用いて文対応を取るといったコストのかかる処理が必要となる。

本稿では、大量の日韓対訳文書から、対訳文対コーパスを生成することなしに、直接文字間の対応を学習する手法を提案する。実験により、対訳文書のみから学習した文字ベースの翻訳モデルで、高性能な日韓・韓日統計的機械翻訳システムが構築可能であることを示す。

2　日本語と朝鮮語の類似性

2.1　語彙

日本語と朝鮮語の語彙には大量の漢語と外来語が含まれており、これらの日韓語彙間では文字単位の対応ができる。また、助詞などの構文上で重要な形態素においても両言語間である程度の対応が存在している。

図 1: 日本語と朝鮮語の文字単位の対応例。朝鮮語側の空箱は正書法上のスペースを示す。

図 1 の例において、「米国」、「戦略」は漢語であり、厳密な翻字が可能である。「グローバル」のように音訳された外来語も、文字単位の対応が可能である。さらに、「の」、「を」のような構文上重要な形態素も対応している。

2.2　構文

日本語と朝鮮語は共に典型的な膠着語である。具体的には、名詞のような「体言」の後ろに助詞が接続し、文における役割や強調などの情報を示す。また、動詞のような「用言」の後ろに助動詞が接続し、態・相・時制などの情報を示す。被修飾部分は常に修飾部分の後ろに、動詞は常にその支配する名詞の後ろに来る。すなわち、構文上では両言語ともに「head-final」な言語であり、語順も一致度が高い（図 2）。

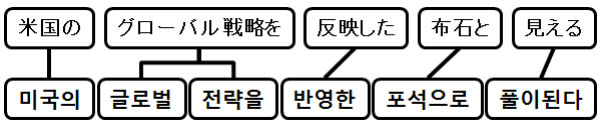

図 2: 日本語と朝鮮語のフレーズ単位の対応例。日本語側の箱は文節を、朝鮮語側の箱は正書法上の分かち書きを示す。

3　対訳文書からのモデル学習

前述した語彙及び構文の一致性により、日本語・朝鮮語の間では、「文」や「形態素」などの概念を意識せずに、対訳文書全体で、モノトーンな文字対応が可能であるという仮説を立てた。

対訳文対間の単語対応を学習する手法として、IBM モデルと HMM モデルが挙げられる [2]。IBM モデル 1 は単語の共起のみに着目し、単語間の対訳確率を推定するモデルである。IBM モデル 2 と HMM モデル

は単語間の位置関係も考慮したモデルであり、IBM モデル 2 は「絶対位置」を推定し、HMM モデルは「相対位置」を推定する。IBM モデル 3 以上のモデルは単語間の一対多といった現象を反映する「繁殖率」も考慮に入れる。しかし、単語対応の対称化のヒューリスティック [2, 3] を利用する場合、IBM モデル 3 以上の複雑なモデルを利用しても Phrase-based 翻訳システムの性能の著しい向上が見られないことが示されている [3]。日本語・朝鮮語間の類似性と IBM モデルの特性により、日本語・朝鮮語間の文字単位の対応は IBM モデル 1 でも対応できると考えた。

アルゴリズム 1 は IBM モデル 1 のパラメータの推定手法を示している[1]。このアルゴリズムは、それぞれの対訳文対に対して（5 行目からの **for** 文の内部）、$O(|\mathbf{e}|\cdot|\mathbf{f}|)$ の計算量を必要とする（6 行目と 8 行目、10 行目と 11 行目の 2 重の **for** 文）。対訳文対の長さは線形の関係を仮定し、ともに n に比例する単語を含むとすると、アルゴリズム 1 のコアとなる部分の計算量は $O(n^2)$ となる。文書は文の長さよりもはるかに長いため、二乗の計算量となる IBM モデル 1 の推定アルゴリズムを、対訳文書対にそのまま適用するのは非現実的である。

IBM モデル 1 は対訳文対の単語が位置にかかわらず対応することを仮定しており、すべての組み合わせを考慮しなければならない（図 3 の左）。これに対して対訳文書対の場合、一方の文書の最初の部分は他方の文書の最後の部分と対応しているとは考えにくいことから、本稿ではアルゴリズム 1 に「対応範囲を限定する」というヒューリスティックを適用した（図 3 の右）。具体的には、アルゴリズム 1 における「**for** all words f in $\mathbf{f}$ **do**」の部分（8 行目と 11 行目）を、「**for** all words f in **beam do**」と変更する。**beam** の幅 b を $f(|\mathbf{f}|)$ のような $|\mathbf{f}|$ の関数にすると、前述した $O(n^2)$ の計算量は $O(n\cdot f(n))$ となる。すなわち $o(n)$ の $f(n)$ を設定することにより計算量の削減が実現できる。

Algorithm 1 EM for IBM model 1 (Fig. 4.3 in [4])

Require: set of sentence pairs $(\mathbf{e}, \mathbf{f})$

1: initialize translation prob. $t(e|f)$ uniformly
2: **while** not converged **do**
3: $count(e|f) = 0$ for all e, f
4: $total(f) = 0$ for all f
5: **for** all sentence pairs $(\mathbf{e}, \mathbf{f})$ **do**
6: **for** all words e in $\mathbf{e}$ **do**
7: $s\text{-}total(e) = 0$
8: **for** all words f in $\mathbf{f}$ **do**
9: $s\text{-}total(e) += t(e|f)$
10: **for** all words e in $\mathbf{e}$ **do**
11: **for** all words f in $\mathbf{f}$ **do**
12: $count(e|f) += \frac{t(e|f)}{s\text{-}total(e)}$
13: $total(f) += \frac{t(e|f)}{s\text{-}total(e)}$
14: **for** all f **do**
15: **for** all e **do**
16: $t(e|f) = \frac{count(e|f)}{total(f)}$
17: **return** translation prob. $t(e|f)$

[1]本稿では [4] の Figure 4.3 の疑似コードを参照し提案手法を説明する。ただし、[4] と [2] とでは「e」と「f」が逆になっている。

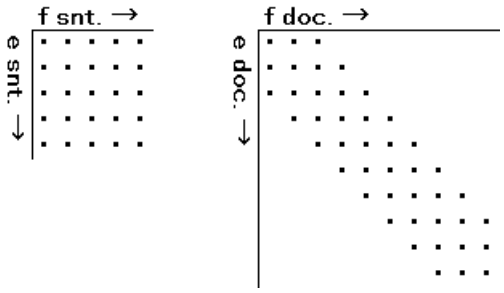

図 3: 従来の IBM モデル 1 と提案手法。「・」は推定に計算すべき単語対である。左は $|\mathbf{e}|\cdot|\mathbf{f}|$ 個の単語対を考慮する従来の IBM モデル 1 である。右は $|\mathbf{e}|\cdot f(|\mathbf{f}|)$ 個の単語対を考慮する提案手法である。

4 評価実験

4.1 対訳データ

東亜日報のウェブサイト[2]から日韓対訳記事の本文を収集し、提案した対訳文書対からのモデル学習手法を評価する。この対訳記事データから抽出した対訳文対で学習したモデルをベースラインとし、提案手法による劣化の程度を評価する。対訳文対の抽出には、類似度の計算に日本語・朝鮮語辞書を利用した DP マッチングの手法 [5] を用いる。

第 2.1 小節で述べた日本語・朝鮮語間の文字単位対応の可能性から、記事データに対する形態素解析を行わず、文字間にスペースを挿入する処理のみを行う。朝鮮語側の正書法上のスペースは 1 文字扱いとし、<sp> というタグで書き換える。記事における段落の区切りも<p>というタグで表記し、1 文字にする。

[2]http://japanese.donga.com/

表 1 は、実験用コーパスの詳細である。train(snt.) はベースラインとする対訳文対コーパス（約 70 万文対）であり、train(doc.) は対訳文書対（約 6 万弱対）である。DP マッチングによって対応が取れない文が存在することから、train(doc.) に対して train(snt.) の文字数が減少している。デベロップメント (dev.(doc.)) 及びテスト (test(doc.)) には記事文書そのものを使う。

表 1: コーパスの詳細

コーパス	期間	数	文字数 (日/韓)
train(snt.)	2001.2-2013.5	701,912	$41.3M/43.1M$
train(doc.)	2001.2-2013.5	57,551	$44.9M/47.4M$
dev.(doc.)	2013.6-2013.7	558	$531K/583K$
test(doc.)	2013.8-2013.9	549	$531K/583K$

4.2 モデルの学習

ベースラインは train(snt.) に対して GIZA++[3]を使用し、提案手法は train(doc.) に対して前述したヒューリスティック付きの IBM モデル 1 を利用して[4]文字対応を学習する。提案手法の学習における対応制限幅 b を $|\mathbf{f}|^{0.5}$ とした。具体的には、$\mathbf{e}$ の i 番目の文字に対して、$\mathbf{f}$ における

$$[\lfloor i\frac{|\mathbf{f}|}{|\mathbf{e}|}-\frac{b}{2}\rfloor,\lceil i\frac{|\mathbf{f}|}{|\mathbf{e}|}+\frac{b}{2}\rceil]\cap[0,|\mathbf{f}|]$$

の範囲を **beam** にする。文字対応の学習は両方向で行い、図 4 は提案手法の反復回数に従った train(doc.) 上の文字当たりのパープレキシティーである。

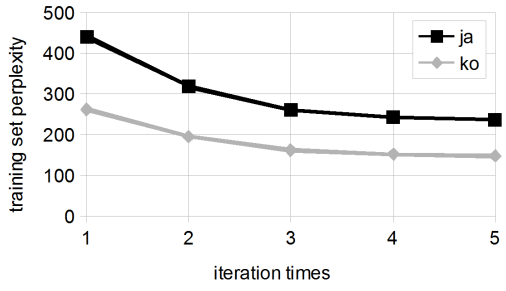

図 4: train(doc.) 上のパープレキシティー（文字あたり、ja は日本語、ko は朝鮮語）

[3] http://code.google.com/p/giza-pp/
[4] 5 回反復

Moses[5]に含まれる単語対称化ツールを利用し、「grow-diag-final-and」で文字対応の対称化を行った[6]。Moses のスクリプトを利用し、Phrase-based 翻訳モデルと Lexicalized orientation reordering モデルを作成した。フレーズの最大文字数を 9 に設定した。また、train(doc.) 上で SRILM[7]を利用して Interpolated Modified Kneser-Ney 法で文字 9-gram を作成した。この言語モデルはベースライン・提案手法両方の実験に使う。

第 2.2 小節で述べた日本語・朝鮮語間の語順一致性により、デコーディングにおける distortion-limit を 0、ttable-limit を 10、stack を 20[8]とし、「drop-unknown」オプションで未知語を無視する[9]。なお、train(snt.) は段落分け記号<p>を含まないため、train(snt.) で学習した翻訳モデルに以下のルールを追加する。

<p> ||| <p> ||| 1.0 1.0 1.0 1.0 2.718 |||

4.3 結果と考察

図 5 と図 6 は対訳文対・対訳文書対コーパス上で [6] の手法により算出した kendall's τ である。第 2.2 小節で述べた日本語・朝鮮語間の語順の一致性を裏付けている。

図 6 に示す kendall's τ の分布は図 5 よりも 1.0 付近に集中している。これは「幅」を設けることで、強制的にほぼモノトーンな文字対応を生成するからである。なお、図 5 と図 6 の分布が似ていることから、提案手法で用いたヒューリスティックは合理的であると言える。

表 2 と表 3 にテストセット上の翻訳性能を示す。デコーディングにおける distortion-limit を 0 （語順入れ替え無し）にしたため、Lexicalized orientation reordering モデルの有無による比較実験も行った。*char* は文字単位の評価結果であり、*moprh* はシステムの日本語出力を MeCab[10]で形態素解析を行い、その出力をもとに計算した評価結果である。*ortho* は朝鮮語の正書法による分かち書きに復元した[11]単位による評価結果である。

[5] http://www.statmt.org/moses/
[6] Moses の symal という単語対称化ツールを利用した。ただし、長い文書対にも使えるように、symal.cpp における最大単語数を定義するマクロを修正した。
[7] http://www.speech.sri.com/projects/srilm/
[8] この設定で、デコーディング時のメモリ使用量は約 20–30G になる。
[9] ここでの「未知語」はすなわち未知な文字である。
[10] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
[11] 文字間のスペースを削除し、前述した<sp>タグは文字として翻訳されるので、システムの出力に<sp>をスペースに置換する。

ベースラインである対訳文対の学習では、韓日（ko-ja）・日韓（ja-ko）翻訳共に比較的高い BLEU 値を達成している。Lexicalized orientation reordering モデルは、システムの性能にあまり影響しないことが分かった。提案手法による学習は、目的言語が日本語の場合、ベースラインとの性能差がわずかである。同じように、Lexicalized orientation reordering モデルは性能に影響がない。一方、目的言語が朝鮮語の場合、提案手法とベースラインの性能差が大きくなっている。Lexicalized orientation reordering モデルを利用しない場合、性能の悪化が見られる。

翻訳文の語順並べ替えを反映した評価指標である RIBES は、表 3 で示すように手法などにかかわらず、非常に高い数値になっている。これは図 5・図 6 で示した語順の一致性から順当に得られた結果であると考える。

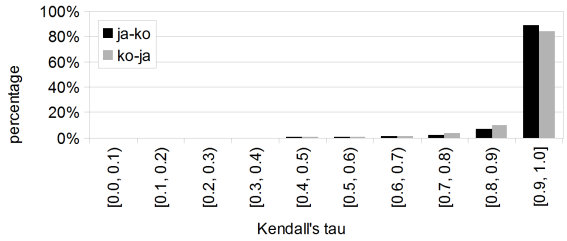

図 5: 対訳文対コーパスから GIZA++で学習した文字対応の Kendall's τ。（ja-ko と ko-ja は [6] と同じ意味）

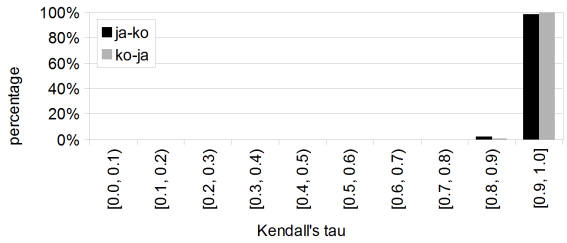

図 6: 対訳文書対コーパスから提案手法で学習した文字対応の Kendall's τ。（ja-ko と ko-ja は図 5 と同じ）

5 おわりに

本稿では、日本語と朝鮮語といった類似した言語間で、対訳文書対上で直接文字対応を学習する手法を提案した。具体的には、単語共起に注目する従来の IBM モデル 1 の推定アルゴリズムにヒューリスティックを加えることにより、対訳文書対にも適用できるようにした。実験により、日本語と朝鮮語間は、対訳辞書で

表 2: 各翻訳モデルのテストセット BLEU。snt. は対訳文対から、doc. は対訳文書対から学習した翻訳モデル。左・右の数値は Lexicalized orientation reordering モデル使用・不使用の結果。

	ko-ja$_{char}$	ja-ko$_{char}$	ko-ja$_{morph}$	ja-ko$_{ortho}$
snt.	61.9/62.2	72.1/71.9	52.2/52.0	35.1/34.6
doc.	59.2/59.3	68.7/67.9	48.5/48.6	30.3/29.2

表 3: 各翻訳モデルのテストセット RIBES。記号の意味は表 2 と同じ。

	ko-ja$_{char}$	ja-ko$_{char}$	ko-ja$_{morph}$	ja-ko$_{ortho}$
snt.	.883/.878	.909/.907	.880/.874	.858/.856
doc.	.864/.863	.892/.887	.860/.859	.836/.831

対訳文対を取らずとも、比較的大きい単位の対訳データから有効に翻訳モデルを学習できることを示した。

参考文献

[1] David Vilar, Jan-Thorsten Peter, and Hermann Ney. Can we translate letters? In *Proc. of Workshop of SMT*, pp. 33–39, 2007.

[2] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, 2003.

[3] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proc. of NAACL*, pp. 48–54, 2003.

[4] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

[5] Masao Utiyama and Hitoshi Isahara. A Japanese-English patent parallel corpus. In *Proc. of MT Summit*, pp. 475–482, 2007.

[6] Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. HPSG-based preprocessing for English-to-Japanese translation. *ACM Transactions on Asian Language Information Processing*, Vol. 11, No. 3, 2012.