モダリティの特徴語を用いたフレーズベース統計的機械翻訳

井手上 雅迪† 山本 和英† 内山 将夫‡ 隅田 英一郎‡
長岡技術科学大学 電気系† 情報通信研究機構 MASTAR プロジェクト‡

1 はじめに

統計的機械翻訳の出力文には訳語選択や語順の間違い等、複数の観点における翻訳誤りが含まれる。また、否定文の入力に対して肯定文を出力してしまうというような、入力文のモダリティが反映されていないという誤りも含まれる。訳語選択や語順の誤りは出力結果の不自然さから人間が容易に発見できると考えられるが、モダリティの翻訳誤りは正しそうな出力結果でも誤っていることがあるため、そのような出力結果の意味を利用者がそのまま受け取ってしまう可能性がある。

本稿では否定・肯定・疑問のモダリティを対象に、各モダリティの特徴語を考慮した素性関数をフレーズベース統計的機械翻の翻訳モデルに組み込む手法を提案する。また、各モダリティの特徴語を考慮することで入力文のモダリティを保存した翻訳ができることを翻訳実験から示し、特徴語の抽出方法についても、人手による手法と自動で抽出する手法を比較する。

2 関連研究

統計的機械翻訳でモダリティの保存を考慮した研究に Finch et al.[1] と Goh et al.[3] がある。Finch et al. の研究では、まず通常の翻訳モデルと疑問文の翻訳モデル、それ以外の翻訳モデルを用意する。次に文頭と文末を表すタグも含めた n-gram を素性とした最大エントロピー分類器により、原言語文から出力結果が疑問文、またはそれ以外の文であるかを分類する。その結果から各モデルに対する重みを変更する。分類精度は複数言語間において 90% 以上となっており、翻訳精度も向上している。

Goh et al. の研究は統計的機械翻訳の翻訳候補をリランキングすることで翻訳精度を向上させることを目的としている。Support vector machine ベースのリランキングには様々な素性が用いられ、Finch et al. と同様に、疑問文か叙述文か推定した結果も用いられている。また、疑問文か叙述文かだけではなく、否定文か肯定文かの推定も行なっている。

Finch et al. と Goh et al. の研究では翻訳精度が向上しているが、どのような表現が各モダリティに影響を与えるのか議論されていない。本稿では肯定・否定・疑問の各モダリティにおける特徴語に着目し、特徴語を考慮した素性関数を既存の翻訳モデルに追加することで入力文のモダリティを保存した翻訳を目指す。更に、人手によって抽出された特徴語と自動抽出された特徴語について比較を行う。

†{ideue,yamamoto}@jnlp.org
‡{mutiyama,eiichiro.sumita}@nict.go.jp

表 1: 人手抽出による特徴語 (英語)

否定	not	No	cannot	't
疑問	?	Why	Will	What
	Could	Is	How	Does
	Can	Do	Are	Which
	When	Where	Have	Does
	Did	Was	May	Shall

3 モダリティの特徴語を考慮した翻訳

否定と疑問のモダリティにおいて、特徴語の有無を考慮した素性関数を提案する。この素性関数をフレーズベース統計的機械翻訳の翻訳モデルに組み込むことで、入力文のモダリティを保存した日英翻訳を行う。

否定と疑問のモダリティにおける特徴語とは、各モダリティを表す可能性の高い語のことである。表 1 に人手で設定した英語の特徴語を示す。be 動詞など文頭に位置することで疑問を表す語のように、モダリティを表す特徴語は文内での位置にも依存することがある。そのため、特徴語の大文字と小文字を区別する。

英語の特徴語は、統計的機械翻訳のツールキットである Moses[4] に付属のトークナイザによってトークン化したものを語の単位として扱う。

3.1 翻訳モデル

標準的なフレーズベース統計的階翻訳では以下の対数線形モデルに基づいて出力文 $\hat{\boldsymbol{e}}$ が決定される。

$$\hat{\boldsymbol{e}} = \arg\max_{\boldsymbol{e}} \sum_{i=0} \lambda_i h_i(\boldsymbol{e}, \boldsymbol{f}, \boldsymbol{c}) \tag{1}$$

ここで f は入力された日本語文であり、$\boldsymbol{e}$ は $\boldsymbol{f}$ に対する英語文の仮説である。$\boldsymbol{c}$ は各言語のフレーズ間の対応関係を表す。$h_i(\boldsymbol{e}, \boldsymbol{f}, \boldsymbol{c})$ は素性関数であり、言語モデルやフレーズ翻訳確率等を用いる。λ は各素性関数に対する重みである。

3.2 モダリティを考慮した素性関数

モダリティを考慮した素性関数を以下に定義する。

$$\phi_m(\boldsymbol{e}) = \sum_i f_c(\bar{e}_i) \tag{2}$$

$$f_c(\bar{e}) = \begin{cases} 1 & if\ |\boldsymbol{C}_{\bar{e}} \cap \boldsymbol{C}| \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

表 2: 人手抽出による特徴語 (日本語)

否定	ない	ません
疑問	?	か。

表 3: 否定の特徴語抽出で使用する分割表

	否定	肯定
$w=1$	a	b
$w=0$	c	d

特徴語集合を $\boldsymbol{C}$ とすると、$f_c(\bar{e})$ はフレーズ $\bar{e}$ の単語集合 $C_{\bar{e}}$ 内に特徴語がひとつ以上含まれていれば 1 となる。(2) 式より、$\phi_m(\boldsymbol{e})$ は特徴語をひとつ以上含む英語側のフレーズ数となる。

(1) 式に組み込む素性関数は、特徴語集合を表 1 の否定側に示した語とする $\phi_{neg}(\bar{e})$ と、疑問側に示した語とする $\phi_{que}(\bar{e})$ である。

3.3 両言語での特徴語

3.2 節では目的言語である英語の仮説のみについて特徴語を考慮しているが、入力文のモダリティを保存するという観点では、原言語である日本語側の特徴語も考慮することが自然である。

そこで (2) 式の f_c を、英語側のフレーズと日本語側のフレーズの両方に特徴語が含まれている場合に 1 となるように変更する。

日本語側の特徴語には表 2 を使用する。翻訳モデル学習の際は、形態素解析器である ChaSen* を用いて日本語文の形態素分割を行うので、表 2 は複数形態素にわたる特徴語を含んでいる。

3.4 Log-Likelihood Ratio による特徴語抽出

表 1 や表 2 に示した特徴語は、否定と疑問の各モダリティを強く表す語として人手で抽出したものであるが、使用するコーパスによっては人手で抽出した特徴語以外にも、特徴語であると認められる語が存在する。例えば旅行会話の場合では、値段を尋ねるような「いくら」という語も特徴語であるとみなすことができる。しかし、このような語も人手で網羅的に抽出することは難しく、多言語に対応させる場合では大きなコストとなる。

そこで、特徴語を自動で抽出することを考える。Chujo et al.[2] は log-likelihood ratio (LLR) を含む様々な統計的尺度を用いて、旅行分野に特化した語を抽出している。本稿では LLR を用いて特徴語を抽出する。はじめに、LLR を計算するために表 3 に示すような分割表を作成する。LLR を計算したい語を w とすると、否定の特徴語抽出を例にしたとき、分割表のパラメータは以下のようになる。

- $\boldsymbol{a}$ w の否定文における出現頻度
- $\boldsymbol{b}$ w の肯定文における出現頻度
- $\boldsymbol{c}$ w 以外の語の否定文における出現頻度
- $\boldsymbol{d}$ w 以外の語の肯定文における出現頻度
- $\boldsymbol{n}$ 全ての語の出現頻度

LLR の定義式を (4) 式に示す。ここで D は表 3 のパラメータ群の状態である。w は否定と肯定のどちらにも依らず独立に出現するという仮説を H_{indep} とし、従属の場合を H_{dep} とする。$sign(ad-bc)$ はどちらのモダリティに依存しているか測るための関数である。$ad-bc$ が 0 以上の場合に 1 となり、それ以外では -1 となるので、否定文での w の出現確率が肯定文よりも高い場合に LLR は正の値となる。

$$LLR = sign(ad-bc)LLR_0 \tag{4}$$

$$sign(z) = \begin{cases} +1 & if\ z \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{5}$$

$$LLR_0 = \frac{Pr(D|H_{dep})}{Pr(D|H_{indep})} \tag{6}$$

$$= a\log\frac{an}{(a+b)(a+c))} + b\log\frac{bn}{(a+b)(b+d))} + c\log\frac{cn}{(a+b)(a+c))} + d\log\frac{dn}{(c+d)(b+d))}$$

最後に LLR の降順で各語を順位付けし、上位 N 語を特徴語として抽出する。疑問の特徴語においても同様に抽出する。特徴語抽出は日英両言語に対して行い、両言語を考慮した素性関数を適用する。LLR による日本語側特徴語の単位は 1 形態素としている。

3.5 各モダリティに属する文の抽出

LLR による特徴語抽出では各モダリティにおける語の出現頻度を得るために、コーパス中の文を各モダリティに分類する必要がある。分類は表 1 の否定側の特徴語が対訳コーパスの英語文に含まれていれば否定文、文末が疑問符であれば疑問文、どちらでもない場合は肯定文とする。疑問とも否定とも判定される文は疑問文として扱う。

英語文の場合は人手特徴語による各モダリティの判定が容易であるが、日本語文は英語文と比較して各モダリティの判定が難しい。そのため、日本語文の分類は英語文を基準として行う。ある日本語文を分類する場合、対訳関係にある英語文が分類されるモダリティに日本語文を分類する。

4 実験

翻訳には Moses[4] を用いた。素性関数の重み推定には Minimum Error Rate Training [5] を用いた。対訳コーパスには旅行会話対訳コーパスである Basic Travel Expression Corpus の約 70 万文を使用し、テストデータには 3.5 節の方法で肯定・否定・疑問の文を 500 文ずつ選択して使用した。チューニング用の開発データは dev1 と dev2 の 2 種類を用意し、dev1 は無作為に抽出した 1500 文、dev2 はテストデータと同様に各モダリティの文を 500 文ずつ使用した。dev1 と dev2 は素性追加なしで比較し、開発データのみの

*http://chasen-legacy.sourceforge.jp/

表 4: BLEU による評価

	素性なし (dev1)	素性なし (dev2)	MAN_E	MAN_EJ	LLR20	LLR30	LLR50	LLR100
BLEU	32.61	32.86	32.84	32.73	32.58	32.75	32.57	32.61
	-0.25		-0.02	-0.13	-0.28	-0.11	-0.29	-0.25

表 5: 人手による翻訳品質の評価 (文)

	S	A	B	C	D
ベースライン	60	57	34	26	93
MAN_E	66	40	38	29	97
MAN_EJ	55	54	44	29	88
LLR30	60	56	38	28	88

表 6: 各モダリティの精度 (括弧内は各モダリティでの文数)

	肯定 (135)	否定 (51)	疑問 (84)
ベースライン	86.67	39.22	90.48
MAN_E	71.11	80.39	95.24
MAN_EJ	87.41	64.71	90.48
LLR30	87.41	62.75	95.24

違いで出力結果に影響があるか調べた。素性関数を追加した場合の開発データは dev2 を用いる。学習データは残りの対訳文を使用した。

素性なしの場合と人手による英語特徴語 (MAN_E)、人手による日英特徴語 (MAN_EJ)、LLR による日英特徴語における比較では、翻訳精度と、翻訳結果が入力文のモダリティを保存しているかという点を評価する。

4.1 BLEU による評価

BLEU による翻訳精度の評価を表 4 に示す。LLR に続く数値は上位 N 語を特徴語として抽出したことを表す。全体的に翻訳精度の差は小さいが、素性なし (dev2) の場合が最も BLEU が高く、素性を追加した場合の BLEU に対する効果は見られなかった。

4.2 人手による評価方法

本研究の手法では出力結果の差異が、特徴語を含むか含まないかという数単語での変化になるので BLEU に大きな変化は与えないと考えた。そこで、素性なし (dev2) をベースライン手法とし、MAN_E、MAN_EJ、LLR の出力結果について人手で評価した。LLR については、BLEU が最高であった LLR30 を選択した。

評価するデータは先に述べた 4 手法の出力結果のうち、出力結果が全て同じものを除外し、残りのテストデータから各モダリティ毎に 90 文を無作為に選択した。評価結果を表 5 と表 6 に示す。翻訳品質は D から S までの 5 段階評価で示す。各モダリティの精度は出力文のモダリティを判定し、その何割が入力文のモダリティと一致しているかを示す。

表 5 より、S と A と評価された翻訳を正解とすると、どの実験設定でも 106 文から 117 文が正解であり、BLEU と同様に翻訳品質において大きな差はない。

4.3 各モダリティの精度

表 6 より、素性関数を追加しない場合は否定文の精度が 39.22%と低いが、素性関数を追加すると特徴語集合に関係なく否定文の精度が向上している。以下に本手法を適用することで出力が改善された例を示す。

入力文 サーカスと動物園、どっちに行こうか。
ベースライン Let's go to the circus and, the zoo? (×)
MAN_EJ Which one shall we go to the circus and Zoo? (◯)

上例では入力文が疑問文であるのに対し、ベースライン手法では疑問を表す語が疑問符だけであり、全体としてみると疑問文ではない。(MAN_EJ) の出力では、疑問符だけでなく Which も疑問を表す語として含まれている。特徴語を考慮した素性関数を追加すると、モダリティの特徴を表す語を含むフレーズが翻訳候補に現れた場合、そのようなフレーズを優先的に使用することになるので、モダリティを表す語を多く含むことによって入力文のモダリティが保存される。

MAN_E では、肯定文において精度がベースラインよりも低下する。本手法で (1) 式に追加する素性関数は ϕ_{neg} と ϕ_{que} のみであり、肯定文に対する考慮はなされていないので素性関数の重みが負でない限り、否定もしくは疑問の特徴語を含むフレーズを優先的に使用することになる。英語のみの特徴語では、表 1 で挙げた語はそれぞれ出現頻度が高く多くのフレーズペアに含まれるので、日本語側フレーズのモダリティに関係なく、英語側が特徴語を含んでいる不適当なフレーズを選択するようになる。MAN_EJ では肯定文の精度は低下していない。MAN_EJ は入力文にも特徴語を含む必要があるという制限があるので、英語側特徴語の影響で不適当なフレーズを選択することを防ぐ。

疑問文の精度は素性関数の追加なしでも 90.48%と高いが、英語のみの人手による特徴語と LLR30 では精度が更に向上する。

素性なしと比較して、肯定文、否定文、疑問文の全てで精度が向上しているのは LLR30 のみであり、特徴語を自動抽出しても翻訳精度を維持したまま各モダリティの精度が向上している。

4.4 LLR による特徴語

表 7 に LLR によって抽出した特徴語の例を示す。LLR による英語側の特徴語は、人手で抽出した特徴語を全て含んでいる。表 7 は人手による特徴語を除去した語の上位 15 個を抽出したものである。また英語の否定側は didn など 't の前に接続する語も除去して表示している。各モダリティでの精度において LLR30 は人手 (日英) と同等以上の結果となっているので、特

表 7: LLR によって抽出した特徴語

否定		疑問	
英語	日本語	英語	日本語
can	ませ	do	か
yet	ない	't	どこ
any	ん	any	何
but	は	there	どう
know	なかっ	have	いくら
worry	あまり	this	は
I	まだ	don	どの
anything	あり	long	いただけ
it	でき	it	何時
so	じゃ	isn	あり
afraid	いいえ	did	もらえ
understand	そんなに	your	でしょ
what	わから	much	いかが
enough	そんな	how	ます
work	たく	time	どんな

徴語として適当なものが抽出されていると考えられるが、実際には抽出された特徴語には"much" や "あり" 等の、1 語ではモダリティを判別できない語が含まれる。このような適当でない特徴語を入力文に含むだけで、別のモダリティに翻訳されることがある。以下に" あり" や" ます" が含まれているだけで肯定文の入力に対し疑問文の特徴語を含んだ出力となった例を示す。

(入力文) 年に一度昇給を得る資格があります。
(LLR30) Do you have any qualifications do you get a raise once a year.

今回は日本語側の人手特徴語を "か 。" のように複数形態素で扱うことで、モダリティを強く表すような特徴語にしている。上記のような翻訳誤りを防ぐためには、単語バイグラムや単語トライグラムなど、複数語の単位で特徴語を扱うことを検討する必要がある。

表 6 の括弧内の数値は入力文の各モダリティにおける文数であり、人手で確認している。人手評価用のデータは 3.5 節の方法により肯定文・否定文・疑問文に 90 文ずつ分割しているが、実際は 90 文あるはずの否定文が 51 文しか存在していないなど、分類性能が高くないことが分かる。

LLR によって、ノイズを含むコーパスから抽出した特徴語でも人手による特徴語と同等以上の性能がある。特徴語として上位からいくつの語を抽出すれば良いのか適切に設定できれば、LLR による特徴語抽出には頑健性があるといえる。

4.5 翻訳失敗例

本手法による翻訳失敗例を以下に示す。

(入力文) やさしく打ってくださいね。
(MAN_E) Please go easy, isn't it? (×)
(MAN_EJ) Please go easy. (○)

上記は MAN_ E で "〜, isn't it?" と翻訳され、付加疑問文となった例である。"isn't it?" という表現は疑問文と否定文の特徴語を含んでいるため、積極的に翻訳される傾向にある。MAN_ EJ では選択するフレーズペアを日本語の特徴語で制限しているので正しい翻訳となっている。

(入力文) キャンセルしてもかまいませんか。
(ベースライン) May I cancel? (○)
(MAN_EJ) I don't mind if you cancel it? (×)

上記は入力が疑問文であるのに対し、MAN_EJ の出力が否定文として判定された例である。これは入力文に否定文特徴語の" ません" と疑問文特徴語の" か。"が含まれているためである。"ません" は否定のモダリティを強く表す語として人手により抽出されたが、例のように常に否定を表すとは限らない。

5 おわりに

本稿では、肯定・否定・疑問のモダリティを保存した日英翻訳を実現するために、各モダリティの特徴語を考慮した素性関数をフレーズベース統計的機械翻訳の翻訳モデルに組み込む手法を提案し、実験を行った。

本手法を適用すると翻訳精度を維持したまま入力と出力のモダリティの一致数が増える結果となった。特に、本手法は否定のモダリティに対して有効であった。

日英両言語の特徴語を人手抽出した場合と LLR によって自動抽出した場合を比較すると、特徴語を自動抽出した場合でも人手抽出と同等以上の性能となった。

参考文献

[1] Finch Andrew, Eiichiro Sumita, and Satoshi Nakamura. Class-dependent modeling for dialog translation. *IEICE TRANSACTION on Information and Systems*, E92-D(12):2469–2477, 2009.

[2] Kiyomi Chujo, Masao Utiyama, and Kathryn Oghigian. Selecting level-specific kyoto tourism vocabulary using statistical measures. *New Aspects of English Language Teaching and Learning*, pages 126–138, 2006.

[3] Chooi-Ling Goh, Taro Watanabe, Andrew Finch, and Eiichiro Sumita. Discriminative reranking for smt using various global features. In *In Proceedings of 4th International Universal Communication Symposium (IUCS 2010)*, pages 8–14, 2010.

[4] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics(ACL)*, pages 177–180, 2007.

[5] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics(ACL)*, 2003.