

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4006297号
(P4006297)

(45) 発行日 平成19年11月14日(2007.11.14)

(24) 登録日 平成19年8月31日(2007.8.31)

(51) Int. Cl.

G06F 17/28 (2006.01)

F I

G06F 17/28

Q

請求項の数 1 (全 16 頁)

(21) 出願番号	特願2002-243908 (P2002-243908)	(73) 特許権者	393031586
(22) 出願日	平成14年8月23日(2002.8.23)		株式会社国際電気通信基礎技術研究所
(65) 公開番号	特開2004-86343 (P2004-86343A)		京都府相楽郡精華町光台二丁目2番地2
(43) 公開日	平成16年3月18日(2004.3.18)	(74) 代理人	100086391
審査請求日	平成15年10月2日(2003.10.2)		弁理士 香山 秀幸
審査番号	不服2004-19261 (P2004-19261/J1)	(72) 発明者	渡辺 太郎
審査請求日	平成16年9月16日(2004.9.16)		京都府相楽郡精華町光台二丁目2番地2
		(72) 発明者	隅田 英一郎
			株式会社国際電気通信基礎技術研究所内
			京都府相楽郡精華町光台二丁目2番地2
			株式会社国際電気通信基礎技術研究所内
		合議体	
		審判長	田口 英雄
		審判官	手島 聖治
		審判官	野崎 大進

(54) 【発明の名称】 統計的機械翻訳装置

(57) 【特許請求の範囲】

【請求項1】

部分出力文 W_T 、出力文の各単語が入力文のどの単語のインデックスに対応するかを示すアライメント A 、既に処理された入力文のインデックスの集合 C 、スコア s 、および open な仮説と close な仮説との識別子 open/ close を要素とする仮説 $h = (W_T, A, C, s, \text{open/ close})$ と、入力文の単語のインデックスの各部分集合に対応した複数のスタックとを用いてデコーディングを行なう統計的機械翻訳装置において、

入力文を入力させる第1手段、

部分出力文 W_T が存在せずかつ識別子 open/ close が close を表す値に設定された初期仮説を生成して空集合に対応するスタックに格納するとともに、変数 c を 0 に設定する第2手段、

要素数が c 個である部分集合に対応するスタックでまだ選択されていないスタックが存在するか否かを判別する第3手段、

要素数が c 個である部分集合に対応するスタックでまだ選択されていないスタックが存在する場合には、そのうちの1つのスタックを選択する第4手段、

第4手段によって選択したスタック内にまだ読み出されていない仮説が存在するか否かを判別する第5手段、

第5手段において、選択したスタック内にまだ読み出されていない仮説が存在すると判別した場合には、まだ読み出されていない仮説を1つ読み出して仮説生成処理を実行した後、第5手段による処理に戻る第6手段、

10

20

第5手段において、選択したスタック内にまだ読み出されていない仮説が存在しないと判別した場合には、第3手段による処理に戻る第7手段、ならびに

第3手段において、要素数が c 個である部分集合に対応するスタックでまだ選択されていないスタックが存在しないと判別した場合には、変数 c を1だけインクリメントし、インクリメント後の c の値が入力文の単語数に達した場合には、今回のデコーディングを終了させ、インクリメント後の c の値が入力文の単語数に達していない場合には、第3手段による処理に戻る第8手段を備えており、

第6手段によって実行される仮説生成処理は、

第6手段によって読み出された仮説 h を入力するとともに、変数 j に1を設定する第1ステップ、

変数 j が入力文の単語数より大きいかなかを判別し、変数 j が入力文の単語数より大きい場合には今回の仮説生成処理を終了し、変数 j が入力文の単語数以下の場合には変数 j が入力仮説 h 内の C に含まれていないかなかを判別する第2ステップ、

第2ステップにおいて、変数 j が入力文の単語数以下であり、かつ変数 j が入力仮説 h 内の C に含まれていないと判別した場合には、入力仮説 h 内の識別子 $open/ close$ に対応した仮説生成処理を実行し、生成した仮説をその仮説中の C に対応するスタックに格納する第3ステップ、ならびに

第2ステップにおいて、変数 j が入力文の単語数以下であり、かつ変数 j が入力仮説 h 内の C に含まれていると判別した場合および第3ステップの処理が終了した場合には、変数 j を1だけインクリメントした後、第2ステップに戻る第4ステップを備えており、

第3ステップは、入力仮説 h 内の識別子 $open/ close$ が $close$ を表す値である場合には、既に得られた部分出力文に対して単語列を追加する操作を行って $close$ に対応した仮説を生成するステップ、および入力仮説 h 内の識別子 $open/ close$ が $open$ を表す値である場合には、入力文の単語を既に得られた部分出力文のある単語に対応付ける操作を行なって $open$ に対応した仮説を生成するステップを備えており、

$close$ に対応した仮説を生成するステップにおいては、出力文を文末から生成するために、入力仮説 h 内の部分出力文 W_T の左側に新しい文字列を挿入することにより新しい部分出力文を生成するとともに入力文のインデックス j にある単語を新しく生成した部分出力文における最初の単語に対応付けし、 $open$ に対応した仮説を生成するステップでは、出力文を文末から生成するために、入力文のインデックス j にある単語を入力仮説 h 内の部分出力文 W_T における最初の単語に対応付けることを特徴とする統計的機械翻訳装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、統計的機械翻訳装置に関する。

【0002】

【従来の技術】

近年、コーパスを用いた手法の一つである統計的機械翻訳システムの研究が盛んに行われるようになってきた。この統計的機械翻訳においては、原言語が与えられたときに目的言語へと翻訳する問題を、最大尤度の解を発見する問題として扱われるようになった。この問題はベーズルールにより最大事後確率の問題としてとらえられ、統計的機械翻訳システムは、翻訳モデルというチャンネルソース言語をチャンネルターゲット言語へと翻訳する確率、チャンネルソース言語の尤度を表した言語モデル、さらに、入力文が与えられたときに出力を発見するデコーダの3つの要素で構成されるようになった(文献1参照)。

【0003】

文献1: Brown, P.F., Pietra, S.A.D., V.L.D. and Mercer, R.L.: The Mathematics of Statistical Machine Translation: Parameter Estimation, Computational Linguistics, Vol.19, No.2, pp.263-311(1993).

【0004】

翻訳モデルでは、確率値を推定する効率的なアルゴリズムが知られているが、デコーディ

10

20

30

40

50

ングはNP_Completeであることが知られている(文献2参照)。

【0005】

文献2: Knight, K.: Decoding Complexity in Word-Replacement Translation Models, Computational Linguistics, Vol.25, No.4, pp.607-615(1999).

【0006】

デコーディングアルゴリズムとして、現在まで様々なアルゴリズム、例えばスタックデコーディングアルゴリズム(文献3参照)、A*探索アルゴリズム(文献4, 5参照)、DPアルゴリズム(文献6, 7参照)が提唱されてきた。

【0007】

文献3: Berger, A., Brown, P., Pietra, S., Pietra, V., Gillett, J., Kehler, A. and Mercer, R.: Language Translation Apparatus and Method of Using Context-Based Translation Models, Technical report, United States Patent, Patent Number 5510981(1996).

文献4: Och, F.J., Ueffing, N. and Ney, H.: An Efficient A* Search Algorithm for Statistical Machine Translation, Proc. of the ACL-2001 Workshop on Data-Driven Machine Translation, Toulouse, France, pp.55-62(2001).

文献5: Wang, Y.-Y. and Waibel, A.: Decoding Algorithm in Statistical Machine Translation, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics(1997).

文献6: Garcia-Varea, I. and Casacuberta, F.: Search Algorithms for Statistical Machine Translation Based on Dynamic Programming and Pruning Techniques, MT Summit VIII, Santiago de Compostela, Galicia, Spain (2001).

文献7: Tillmann, C. and Ney, H.: Word Re-ordering and DP-based Search in Statistical Machine Translation, Proc. of the COLING 2000(2000).

【0008】

これらは、入力文に対して出力文を文頭から生成し、原言語文と目的言語文とがほぼ線形に対応づけられることを仮定したブルーニング戦略を取り入れたものであった。このようなアルゴリズムにおいては、例えば日本語と英語のように、性質の異なる言語の翻訳には対応できなかった。

【0009】

また、Germannらはヒルクライミングアルゴリズムとインテグラー解を示している(文献8参照)。前者の手法では、満足な解が得られないことが分かっており、また、後者の手法では現実的に処理速度の問題で、解を得ることができない。

【0010】

文献8: German, U., Jahr, M., Knight, K., Marcu, D. and Yamada, K.: FastDecoding and Optimal Decoding for Machine Translation, Proc. of ACL-01, Toulouse, France(2001).

【0011】

この発明は、文末に語順の制約が強いという性質をもつ言語が目的言語である場合に、好適な翻訳結果が得られる統計的機械翻訳装置を提供することを目的とする。

【0012】

【課題を解決するための手段】

この発明による統計的機械翻訳装置は、部分出力文 W_T 、出力文の各単語が入力文のどの単語のインデックスに対応するかを示すアライメントA、既に処理された入力文のインデックスの集合C、スコアs、およびopenな仮説とcloseな仮説との識別子open/closeを要素とする仮説 $h = (W_T, A, C, s, \text{open/close})$ と、入力文の単語のインデックスの各部分集合に対応した複数のスタックとを用いてデコーディングを行なう統計的機械翻訳装置において、

入力文を入力させる第1手段、

部分出力文 W_T が存在せずかつ識別子open/closeがcloseを表す値に設定された初期

10

20

30

40

50

仮説を生成して空集合に対応するスタックに格納するとともに、変数 c を 0 に設定する第 2 手段、

要素数が c 個である部分集合に対応するスタックでまだ選択されていないスタックが存在するか否かを判別する第 3 手段、

要素数が c 個である部分集合に対応するスタックでまだ選択されていないスタックが存在する場合には、そのうちの 1 つのスタックを選択する第 4 手段、

第 4 手段によって選択したスタック内にまだ読み出されていない仮説が存在するか否かを判別する第 5 手段、

第 5 手段において、選択したスタック内にまだ読み出されていない仮説が存在すると判別した場合には、まだ読み出されていない仮説を 1 つ読み出して仮説生成処理を実行した後、第 5 手段による処理に戻る第 6 手段、

第 5 手段において、選択したスタック内にまだ読み出されていない仮説が存在しないと判別した場合には、第 3 手段による処理に戻る第 7 手段、ならびに

第 3 手段において、要素数が c 個である部分集合に対応するスタックでまだ選択されていないスタックが存在しないと判別した場合には、変数 c を 1 だけインクリメントし、インクリメント後の c の値が入力文の単語数に達した場合には、今回のデコーディングを終了させ、インクリメント後の c の値が入力文の単語数に達していない場合には、第 3 手段による処理に戻る第 8 手段を備えており、

第 6 手段によって実行される仮説生成処理は、

第 6 手段によって読み出された仮説 h を入力するとともに、変数 j に 1 を設定する第 1 ステップ、

変数 j が入力文の単語数より大きいかが否かを判別し、変数 j が入力文の単語数より大きい場合には今回の仮説生成処理を終了し、変数 j が入力文の単語数以下の場合には変数 j が入力仮説 h 内の C に含まれていないかが否かを判別する第 2 ステップ、

第 2 ステップにおいて、変数 j が入力文の単語数以下であり、かつ変数 j が入力仮説 h 内の C に含まれていないと判別した場合には、入力仮説 h 内の識別子 $open/ close$ に対応した仮説生成処理を実行し、生成した仮説をその仮説中の C に対応するスタックに格納する第 3 ステップ、ならびに

第 2 ステップにおいて、変数 j が入力文の単語数以下であり、かつ変数 j が入力仮説 h 内の C に含まれていると判別した場合および第 3 ステップの処理が終了した場合には、変数 j を 1 だけインクリメントした後、第 2 ステップに戻る第 4 ステップを備えており、

第 3 ステップは、入力仮説 h 内の識別子 $open/ close$ が $close$ を表す値である場合には、既に得られた部分出力文に対して単語列を追加する操作を行って $close$ に対応した仮説を生成するステップ、および入力仮説 h 内の識別子 $open/ close$ が $open$ を表す値である場合には、入力文の単語を既に得られた部分出力文のある単語に対応付ける操作を行なって $open$ に対応した仮説を生成するステップを備えており、

$close$ に対応した仮説を生成するステップにおいては、出力文を文末から生成するために、入力仮説 h 内の部分出力文 W_T の左側に新しい文字列を挿入することにより新しい部分出力文を生成するとともに入力文のインデックス j にある単語を新しく生成した部分出力文における最初の単語に対応付けし、 $open$ に対応した仮説を生成するステップでは、出力文を文末から生成するために、入力文のインデックス j にある単語を入力仮説 h 内の部分出力文 W_T における最初の単語に対応付けることを特徴とする。

【 0 0 1 3 】

【 発明の実施の形態 】

以下、図面を参照して、この発明の実施の形態について説明する。

【 0 0 1 4 】

〔 1 〕 統計的機械翻訳の概要についての説明

【 0 0 1 5 】

統計的機械翻訳においては、ある原言語のテキスト W_s が与えられているときにある目的言語のテキスト W_T へと翻訳する問題を、最大尤度の出力（翻訳結果）を発見する問題と

10

20

30

40

50

してとらえられ、次式(1)のように表現される。

【0016】

【数1】

$$W_T = \underset{W_T}{\operatorname{argmax}} P(W_T | W_S) \quad \cdots(1)$$

【0017】

上記式(1)にベーズルールを適用することにより、上記の問題は、次式(2)に示すように、チャンネルモデルにおける最大事後確率の解を求める問題に置き換えられる。

10

【0018】

【数2】

$$W_T = \underset{W_T}{\operatorname{argmax}} P(W_T)P(W_S | W_T) \quad \cdots(2)$$

【0019】

このため、翻訳とは、図1(a)に示すように「翻訳家の知識 $P(W_T | W_S)$ を基にして原言語のテキスト W_S を翻訳」する問題として扱われるのではなく、図1(b)に示すように「ある確率 $P(W_T)$ で起こりうる目的言語のテキスト W_T が、チャンネルにより、 $(P(W_S | W_T))$ 、 W_S として出力された」と仮定して、目的言語のテキスト W_T を原言語のテキスト W_S から推測する問題として扱われるようになった(上記文献1参照)。

20

【0020】

$P(W_T)$ は言語モデル(事前確立)であり、目的言語のテキスト W_T の尤度を示す。 $P(W_S | W_T)$ は翻訳モデル(事後確立)であり、目的言語のテキスト W_T が原言語のテキスト W_S へと翻訳される確率を表す。また、翻訳モデルに、目的言語のテキスト W_T の各単語が原言語のテキスト W_S のどの単語のインデックスに対応するかを表すアライメント A という概念が導入されることにより、翻訳モデルは、次式(3)のように表される。

30

【0021】

【数3】

$$P(W_S | W_T) = \sum_A P(W_S, A | W_T) \quad \cdots(3)$$

【0022】

図2は、目的言語のテキスト W_T である英語の文 "Could you recommend another hotel" と、原言語のテキスト W_S である日本語文の「他のホテルを紹介していただけですか」とにおける、アライメントの例を示している。例えば、日本語文におけるインデックス1の「他」とインデックス2の「の」が、英語文のインデックス4の "another" へと対応づけられている。

40

【0023】

また、英語文のインデックス0にあるNULLシンボルも単語の一つであり、「ます」や「か」のように、原言語のテキスト W_S である日本語文に対応する単語がない場合に対応づけられている。

【0024】

〔1.1〕翻訳モデルの構成

【0025】

アライメントを介した翻訳モデル $P(W_S, A | W_T)$ はさらに分解され、例えば、次式

50

(4) で示すように、以下のように4つのモデルから構成される(上記文献1参照)。

【0026】

【数4】

$$\begin{aligned}
 P(W_S, A | W_T) &= \prod_i t(W_{SAi} | W_{Ti}) \\
 &\times \prod_i n(\phi_i | W_{Ti}) \\
 &\times d(A | W_T, W_S) \\
 &\times \binom{m - \phi_0}{\phi_0} (1 - p_1)^{m - 2\phi_0} p_1^{\phi_0}
 \end{aligned} \quad \dots(4)$$

10

【0027】

(1) Lexical モデル: $t(W_{SAi} | W_{Ti})$

ある目的言語の単語 W_{Ti} が原言語の単語 W_{SAi} へと翻訳される確率。

【0028】

(2) Fertility モデル: $n(\phi_i | W_{Ti})$

ある目的言語の単語 W_{Ti} が ϕ_i 個の原言語の単語と対応される確率。

20

【0029】

(3) Distortionモデル: $d(A | W_T, W_S)$

W_T と W_S が与えられたときのアライメントが A である確率。歪みモデルをさらに分解した複雑なモデルも定義されている(上記文献1参照)。

【0030】

(4) NULLモデル: 上記式(4)の右辺の最後の項

ある原言語の ϕ_0 個の単語が NULL へと対応づけられる確率。

【0031】

〔1.2〕探索問題

【0032】

統計機械翻訳における探索問題は、入力文 W_S が与えられたときに、翻訳モデル $P(W_S | W_T) = P(W_S, A | W_T)$ と言語モデル $P(W_T)$ とから最大尤度の出力文 W_T を発見する問題として扱われる。ただし、可能な A の空間は、次式(5)で示されるように、非常に大きく、翻訳を決定する場合、その近似値、 $P(W_S | W_T) \approx P(W_S, A | W_T)$ が用いられる。

30

【0033】

【数5】

$$A \text{ の空間} : |W_S|^{||W_T||+1} \quad \dots(5)$$

40

【0034】

統計機械翻訳における探索問題は、単語の単位の翻訳だけでなく、並び替えの問題も含まれていることから NP_Complete であることが知られている(上記文献2参照)。過去の研究においては、出力を文頭から生成する制約を加え、プルーニング(剪定)を行うことによりデコーディングアルゴリズムを実現している(上記文献3~8参照)。ただし、プルーニングも原言語と目的言語とがほぼ線形に対応づけられていることを仮定しており、日本語と英語間の翻訳といった語順の相違が大きい言語間の翻訳を想定していない。

【0035】

以下、出力の方向を考慮したデコーディングアルゴリズムについて説明する。

【0036】

50

〔 2 〕デコーディングアルゴリズム

【 0 0 3 7 〕

本発明によるデコーディングアルゴリズムは、Tillman（上記文献 7 参照）やOch（上記文献 4 参照）らによるビームサーチアルゴリズムに基づいており、また、仮説に適用される操作はBerger（上記文献 3 参照）、Och（上記文献 4 参照）やGermann（上記文献 8 参照）らに基づいている。

【 0 0 3 8 〕

まず、デコーディングアルゴリズムで用いられる仮説およびマルチスタックについて説明する。

【 0 0 3 9 〕

仮説 h は、 $(W_T, A, C, s, \text{open/close})$ のように表現される。仮説の各要素は次のように定義される。

【 0 0 4 0 〕

W_T : 部分出力文

A : 部分的に決定されたアライメント

C : 既に処理された入力文のインデックスの集合

s : スコア

open/close : open な仮説と close な仮説との識別子であり、0 は open な仮説を表し、1 は close な仮説を表す。

【 0 0 4 1 〕

open な仮説では、入力文の単語を既に得られた部分出力文のある単語と対応づけることにより、fertility（出力文の 1 つの単語に対応付けられる入力文の単語の数）を増大させる操作が行われる。close な仮説では、既に得られた部分出力文に対して単語列 W_T' を追加する操作が行われる。この単語列 W_T' は、fertility が 1 以上の単語 1 個と fertility がゼロの単語の列から構成される。

【 0 0 4 2 〕

マルチスタックは、スタックの集合であり、入力文の単語のインデックスの各部分集合（各組み合わせ）に対応したスタックが存在している。例えば、入力文が、インデックス 1 の単語、インデックス 2 の単語およびインデックス 3 の単語の 3 つの単語から構成されている場合には、入力文の単語のインデックスの各部分集合に応じた、 $C = \{ \}$ 、 $C = \{ 1 \}$ 、 $C = \{ 2 \}$ 、 $C = \{ 3 \}$ 、 $C = \{ 1, 2 \}$ 、 $C = \{ 1, 3 \}$ 、 $C = \{ 2, 3 \}$ 、 $C = \{ 1, 2, 3 \}$ の 7 個のスタックが用意される。

【 0 0 4 3 〕

各スタックには、デコーディングアルゴリズムによって生成された仮説が格納される。各スタックにおいては、スコアの高い順に、仮説がソートされる。スタックに格納されている仮説の数が所定の閾値（max-stack-size）に達している場合に、新たな仮説を格納しようとする場合には、それらの仮説の中でスコアの最も低い仮説を捨てることにより、常に max-stack-size の数の仮説を格納するようにしている。

【 0 0 4 4 〕

〔 2 . 1 〕全体フロー

【 0 0 4 5 〕

図 3 は、デコーディングアルゴリズムを示している。

【 0 0 4 6 〕

まず、入力文 $W_s = W_{s_1}, W_{s_2}, \dots, W_{s_{2m}}$ を入力する（ステップ 1）。入力文 W_s の入力長（単語数）は m である。

【 0 0 4 7 〕

次に初期化処理を行う（ステップ 2）。初期化処理においては、初期仮説 h_0 が生成されて、 $C = \{ \}$ のスタックに格納される。初期仮説 h_0 では、 W_T は存在せず（ $W_T = ""$ ）、アライメント A は - 1 に設定され、入力分のインデックスの集合 C は空（ $\{ \}$ ）に設定され、スコア s は 1 . 0 に設定され、open/close 識別子は close を表す " 1 " に設定さ

10

20

30

40

50

れる。したがって、初期仮説 h_0 は、 $(W_T = " ", A = (-1, \dots, -1), C = \{ \}, s = 1, 0, \text{open/close} = 1)$ となる。また、変数 c が 0 に設定される。なお、以下の説明において、 $|C|$ は、入力文の単語のインデックスの各部分集合の要素数を表すものとする。

【0048】

次に、 $|C| = c$ であるスタック（要素数が c 個である部分集合に対応するスタック）でまだ選択されていないスタックが存在するか否かを判別する（ステップ3）。

【0049】

$|C| = c$ であるスタックでまだ選択されていないスタックが存在する場合には、そのうちの1つのスタックを選択する（ステップ4）。そして、選択したスタック内にまだ読み出されていない仮説が存在するか否かを判別する（ステップ5）。選択したスタック内にまだ読み出されていない仮説が存在する場合には、まだ読み出されていない仮説 h を1つだけ読み出す（ステップ6）。そして、仮説生成処理を実行する（ステップ7）。そして、ステップ5に戻る。

10

【0050】

上記ステップ3において、 $|C| = c$ であるスタックでまだ選択されていないスタックが存在しない場合には、変数 c の値を1だけインクリメント（+1）する（ステップ8）。そして、 c が m に達したか否かを判定する（ステップ9）。 c が m に達していない場合（ $c < m$ ）には、ステップ3に戻る。

【0051】

上記ステップ5において、上記ステップ4で選択したスタック内にまだ読み出されていない仮説が存在しない場合には、ステップ3に戻る。

20

【0052】

上記ステップ9において、 c が m に達していると判別した場合（ $c = m$ ）には、今回の処理を終了する。この時点において、 $|C| = m$ であるスタックの中に解が存在することになる。

【0053】

〔2.2〕仮説生成処理手順

【0054】

図4は、図3のステップ7の仮説生成処理手順を示している。

30

【0055】

以下の説明において、 j は入力文 W_s の単語のインデックスを表す変数であり、 i は出力文 W_T の単語のインデックスを表す変数であるとする。

【0056】

上記ステップ6で読み出された仮説 h を入力する（ステップ11）。変数 j として1を設定する（ステップ12）。

【0057】

そして、 j が $(m + 1)$ に達したか否かを判定する（ステップ13）。 j が $(m + 1)$ に達していない場合（ $j < m$ ）には、 j が入力仮説 h 内の C に含まれていないか否かを判定する（ステップ14）。

40

【0058】

j が入力仮説 h 内の C に含まれていない場合には、入力仮説 h 内の open/close 識別子が open(= 0) であるか、close(= 1) であるかを判定する（ステップ15）。

【0059】

入力仮説 h が open である場合には、open に対応した仮説生成処理を行う（ステップ16）。入力仮説 h が close である場合には、close に対応した仮説生成処理を行う（ステップ17）。open に対応した仮説生成処理および close に対応した仮説生成処理の詳細については、後述する。

【0060】

ステップ16またはステップ17によって新たな仮説が生成された場合には、生成された

50

新たな仮説を、その仮説中のCに対応するスタックに格納する(ステップ18)。

【0061】

そして、変数jの値を1だけインクリメント(+1)した後(ステップ19)、ステップ13に移行する。

【0062】

上記ステップ14において、jがCに含まれていると判定した場合には、ステップ19に移行する。

【0063】

上記ステップ13において、jが(m+1)に達していると判定した場合(j=(m+1))には、今回の仮説生成処理は終了する。つまり、図3のステップ5に戻る。

10

【0064】

以下、openに対応した仮説生成処理およびcloseに対応した仮説生成処理について説明する。

【0065】

{2.3} openに対応した仮説生成処理

【0066】

openに対応した仮説生成処理では、以下のような仮説を、openとcloseとの2種類生成する。

【0067】

入力文 W_S のうちのインデックスjにある単語を、インデックスiにある部分出力文の単語に対応させる。

20

【0068】

この実施の形態では、部分出力文を文末から生成する(right-to-leftデコーディング)ため、インデックスiにある出力単語は部分出力文 W_T における最初の単語(i=1)となる。したがって、図5に示すように、入力文 W_S のうちのインデックスjにある単語 W_{S_j} が、入力仮説h内の部分出力文 W_T における最初の単語 W_{T_1} (i=1)に対応づけられる。

【0069】

この場合には、仮説の要素である部分出力文 W_T は変更されず、アライメントAが変更される。つまり、A[j]として、i=1が代入される(A[j]=i(i=1))。また、仮説の要素であるCとして、C={j}が設定される(C=C∪{j})。

30

【0070】

なお、C={j}は、例えば、Cが{}でかつjが1であれば、{}∪{1}={1}となり、例えば、Cが{1}でかつjが2であれば、{1}∪{2}={1,2}となり、例えば、Cが{2,3}でかつjが1であれば、{2,3}∪{1}={1,2,3}となる。

【0071】

仮説の要素であるスコアsは、よく知られているように次式(6)に基づいて算出される。

【0072】

40

【数6】

$$\begin{aligned} s &\leftarrow s \times t(W_S[j] | W_T[i]) \\ s &\leftarrow s \times (n(f | W_T[i]) / n(f-1 | W_T[i])) \quad \dots (6) \\ s &\leftarrow s \times d(A | W_S, W_T) \end{aligned}$$

【0073】

ここで、fを $W_T[i]$ の新しいferrilityとする。また、Distortionモデル(歪みモデル)の確率を再計算する。

50

【 0 0 7 4 】

以上のような要素を持ち、かつopen/close識別子がopen(= 0)である仮説と、open/close識別子がclose(= 1)である仮説とが生成される。

【 0 0 7 5 】

なお、従来のように部分出力文を文頭から生成する(left-to-right デコーディング)においては、インデックス*i*にある出力単語は部分出力文 W_T における最後の単語($i = |W_T|$)となる。したがって、図6に示すように、入力文 W_S のうちのインデックス*j*にある単語 W_{S_j} が、入力仮説内の部分出力文 W_T における最後の単語 W_{T_i} ($i = |W_T|$)に対応づけられる。この結果、 $A[j]$ として、 $i = |W_T|$ が代入される($A[j] = i (= |W_T|)$)。

10

【 0 0 7 6 】

{ 2 . 4 } close に対応した仮説生成処理

【 0 0 7 7 】

close に対応した仮説生成処理では、NULLへ対応された仮説と、文字列の挿入および挿入文字列へ対応づけられた仮説とを生成する。

【 0 0 7 8 】

{ 2 . 4 . 1 } NULLへ対応された仮説

【 0 0 7 9 】

入力文のうち、インデックス*j*にある単語を、NULL(インデックス0にある出力単語)に対応させる。

20

【 0 0 8 0 】

つまり、 $A[j]$ として、0が代入される($A[j] = 0$)。また、仮説の要素である*C*として、 $C = \{j\}$ が設定される($C = \{j\}$)。仮説の要素であるopen/close 識別子はclose(= 1)とする。仮説の要素であるスコア*s*は、よく知られているように次式(7)に基づいて算出される。

【 0 0 8 1 】

【数7】

$$s \leftarrow s \times t(W_S[j] | NULL)$$

$$s \leftarrow s \times p1/p0^2 \quad \dots (7)$$

$$s \leftarrow s \times \binom{m-f}{f} / \binom{m-f+1}{f-1}$$

30

【 0 0 8 2 】

ここで、*f*をNULLの新しいferrility とする。

【 0 0 8 3 】

{ 2 . 4 . 2 } 文字列の挿入および挿入文字列へ対応づけられた仮説

【 0 0 8 4 】

以下のような仮説を、openとclose との2種類生成する。

40

【 0 0 8 5 】

まず、入力仮説内の部分出力文 W_T に新しい文字列 W_T' を挿入する。この実施の形態では、部分出力文を文末から生成する(right-to-left デコーディング)ため、図7に示すように、入力仮説内の部分出力文 W_T の左側に新しい文字列 W_T' を挿入する($W_T = W_T' + W_T$)。

【 0 0 8 6 】

また、入力文 W_S のうち、インデックス*j*にある単語 W_{S_j} を、インデックス*i*にある出力単語に対応させる。この実施の形態では、部分出力文を文末から生成する(right-to-left デコーディング)ため、図7に示すように、インデックス*i*にある出力単語は新しい*W*

50

T (図7の $W_T' + W_T$) における最初の単語 W_{T1}' ($i = 1$) となる。したがって、アライメント $A[j]$ として、 $i = 1$ が代入される ($A[j] = i (= 1)$)。

【0087】

また、仮説の要素である C として、 $C = \{j\}$ が設定される ($C = C = \{j\}$)。仮説の要素であるスコア s は、よく知られているように次式(8)に基づいて算出される。

【0088】

【数8】

$$\begin{aligned} s &\leftarrow s \times t(W_S[j] | W_T[i]) \\ s &\leftarrow s \times n(I | W_T[i]) \\ s &\leftarrow s \times d(A | W_S, W_T) \\ s &\leftarrow s \times P_{lm}(W_T[i] | W_T[i-2], W_T[i-1]) \end{aligned} \quad \dots (8)$$

10

【0089】

P_{lm} は、言語モデルによるスコアであり、もし、 $W_T[i']$ において、 $i' = 0$ の場合、 $W_T[i'] = BOS$ (Beginning of Sentence Marker) となる。

【0090】

以上のような要素を持ち、かつ open/close 識別子が open (= 0) である仮説と、open/close 識別子が close (= 1) である仮説とが生成される。

20

【0091】

なお、新しい文字列 W_T' は複数生成される。そして、新しい文字列毎に、上記のようにして、2つずつ仮説が生成される。新しい文字列 W_T' の生成方法については、上記文献3, 4, 5に説明されている。

【0092】

従来のように部分出力文を文頭から生成する (left-to-right デコーディング) においては、文字列 W_T を挿入する場合、図8に示すように、入力仮説内の部分出力文 W_T の右側に新しい文字列 W_T' を挿入する ($W_T = W_T + W_T'$)。また、インデックス i にある出力単語は新しい W_T (図8の $W_T + W_T'$) における最後の単語 W_{T1}' ($i = |W_T|$) となる。この結果、 $A[j]$ として、 $i = |W_T|$ が代入される ($A[j] = i (= |W_T|)$)。

30

【0093】

{2.5} 入力文 W_S の入力長 m が3の場合の流れについての説明

【0094】

入力文 W_S の入力長 m が3の場合の、デコーディングアルゴリズムの流れについて簡単に説明しておく。

【0095】

まず、 $C = \{\}$ に対応するスタックに、初期仮説 $h_0 = (W_T = "", A = (-1, \dots -1), C = \{\}, s = 1.0, \text{open/close} = 1)$ が格納される (図3のステップ1~2)。

40

【0096】

そして、図3のステップ6において、 $C = \{\}$ に対応するスタックから初期仮説 h_0 が取り出される。そして、図4のステップ17の close に対応した仮説生成処理によって、 $j = 1, 2, 3$ それぞれに対応した新たな仮説が生成され、 $C = \{1\}$ 、 $C = \{2\}$ 、 $C = \{3\}$ それぞれに対応するスタックに、新たな仮説が格納される。

【0097】

次に、 $C = \{1\}$ に対応するスタックから仮説が1つずつ取り出され、各仮説に基づいて、 $j = 2, 3$ それぞれに対応した新たな仮説が生成され、 $C = \{1, 2\}$ 、 $C = \{1, 3\}$ それぞれに対応するスタックに、新たな仮説が格納される。

50

【0098】

同様に、 $C = \{2\}$ に対応するスタックから仮説が1つずつ取り出され、各仮説に基づいて、 $j = 1, 3$ に対応した新たな仮説が生成され、 $C = \{1, 2\}$ 、 $C = \{2, 3\}$ に対応するスタックに、新たな仮説が格納される。

【0099】

同様に、 $C = \{3\}$ に対応するスタックから仮説が1つずつ取り出され、各仮説に基づいて、 $j = 1, 2$ に対応した新たな仮説が生成され、 $C = \{1, 3\}$ 、 $C = \{2, 3\}$ に対応するスタックに、新たな仮説が格納される。

【0100】

次に、 $C = \{1, 2\}$ に対応するスタックから仮説が1つずつ取り出され、各仮説に基づいて、 $j = 3$ に対応した新たな仮説が生成され、 $C = \{1, 2, 3\}$ に対応するスタックに、新たな仮説が格納される。

10

【0101】

次に、 $C = \{1, 3\}$ に対応するスタックから仮説が1つずつ取り出され、各仮説に基づいて、 $j = 2$ に対応した新たな仮説が生成され、 $C = \{1, 2, 3\}$ に対応するスタックに、新たな仮説が格納される。

【0102】

次に、 $C = \{2, 3\}$ に対応するスタックから仮説が1つずつ取り出され、各仮説に基づいて、 $j = 1$ に対応した新たな仮説が生成され、 $C = \{1, 2, 3\}$ に対応するスタックに、新たな仮説が格納される。

20

【0103】

この結果、 $C = \{1, 2, 3\}$ に対応するスタックに、解（翻訳結果）がスコアの高い順番に格納される。

【0104】

〔3〕デコーディング方向による影響

【0105】

従来の left-to-right デコーディングアルゴリズムにおいては、入力単語を処理し、対応する翻訳を単語単位に決定して部分出力文を文頭から生成する。このような挙動により、言語モデルや翻訳モデルは文頭からスコアリングを行うことにより、「良い仮説」と「悪い仮説」とを区別する力が大きくなることが考えられる。

30

【0106】

例えば、英語やドイツ語、フランス語のように文頭からの語順の制約が強い言語を目的言語として翻訳を行った場合、文頭からのスコアリングによる制約が強くなり、ビームサーチのプルーニングを行っても出力の質が落ちることがない。逆に、このような言語に対して本実施の形態のように right-to-left デコーディングを行うと、仮説間のスコアの差が小さく、仮説数が増大し、プルーニングによる影響が大きくなる。

【0107】

日本語のような文末に語順の制約がある場合、そのような言語を目的言語として right-to-left デコーディングを行うとスコアの差が大きくなり、left-to-right に行くとスコアの差が小さくなると考えられる。したがって、文末に語順の制約がある言語が目的言語である場合には、right-to-left にデコーディングを行うことが好ましい。

40

【0108】

〔4〕実験

【0109】

今回行った実験のコーパスは、大規模旅行会話集（文献9参照）から抽出した172,481 の日英の対訳文であり、その統計情報を表1に示す。コーパスから152,183 文からなるトレーニングセット、検証セットを10,148文、さらにテストセット10,150文を取り出した。

【0110】

文献9：Takezawa, T., Sumita, E., Sugaya, F., Yamamoto, H. and Yamamoto, S.: Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations

50

in the Real World, Proc. of LREC 2002, Las Palmas, CanaryIslands, Spain, pp.147-152(2002).

【 0 1 1 1 】

【 表 1 】

	英語	日本語
文数	172,481	
単語数	1,186,620	1,005,080
語彙数	22,801	15,768
平均文長	6.88	5.83
3-gram パープレキシティー	26.16	36.92

10

【 0 1 1 2 】

翻訳モデルは日英、英日方向、どちらもIBM Model-1 からIBM Model-4 までブートストラップしてトレーニングを行い、途中でHMM Model によるトレーニングを行っている（文献 1 0 参照）。また、検証セットによるクロスバリデーションを行いながらトレーニングを繰り返した。IBM Model-4 においては、品詞を単語クラスとして用いた。

【 0 1 1 3 】

文献 1 0 : Och,F.J., Tillmann,C. and Ney,H.: Improved Alignment Models for Statistical Machine Translation, Proceedings of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora, University of Maryland, College Park,MD(1999).

20

【 0 1 1 4 】

トレーニングコーパスのピテルピアライメントから、fertility がゼロとなる単語列を取り出し、頻度が10未満のものを取り除いた。また、テストセットから文長が6、8、10である、それぞれ50文、合計150 の日本語、英語の文を取り出し、上記の2つのアルゴリズムによるデコーディングの実験を行った。

【 0 1 1 5 】

翻訳結果を評価するに当たって、以下の基準を用いた。

30

【 0 1 1 6 】

・単語誤り率(WER): 挿入、削除、置換に対してペナルティを1として計算。

【 0 1 1 7 】

・位置独立単語誤り率(PER): WER では、実際の翻訳に起こりうる並び替えに対してペナルティが大きくなる。従って、PER では位置の情報を無視し、出力単語の集合の正しさの評価を行う（上記文献 4 参照と文献 1 1 参照）。

文献 1 1 : Watanabe,T., Imamura,K. and Sumita,E.: Statistical Machine Translation Based on Hierarchical Phrase Alignment, Proc. of TMI 2002,Keihanna, Japan(2002).

【 0 1 1 8 】

・BLEUスコア: 正解集合の全てに対してN-gramの正解率を計算するものであり、短い出力に対してペナルティを与えている（文献 1 2 参照）。

40

文献 1 2 : Papineni,K., Roukos,S., Ward,T. and Zhu,W.J.: Bleu: a Method for Automatic Evaluation of Machine Translation, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics(ACL),pp.311-318(2002).

【 0 1 1 9 】

・NISTスコア: BLUEと同様なスコアであるが、正解率を正解集合のN-gramに対する平均出力単語情報量として表現している（文献 1 3 参照）。

文献 1 3 : Automatic Evaluation of Machine Translation Quality using N-gram Co-Occurrence Statistics, <http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf>(20

50

02) .

【 0 1 2 0 】

・主観評価(SE): 翻訳結果を人手でA、B、C、Dと4段階にランク付けをする(文献14参照)。

文献14: Sumita, E., Yamada, S., Yamamoto, K., Paul, M., Kashioka, H., Ishikawa, K. and Shirai, S.: Solutions to Problems Inherent in Spoken-language Translation: The ATR-MATRIX Approach, Machine Translation Summit VII, PP.229-235(1999).

【 0 1 2 1 】

・探索誤り率(SER): 翻訳の正さあるいは誤り以外に、各デコーディングアルゴリズムが翻訳モデルと言語モデルから最適解を得られたかどうかを評価した。出力結果のうち、スコアが最も高い出力の集合を正解集合として、誤り率を求めた。

【 0 1 2 2 】

表2は、left-to-right(LtoR) とright-to-left(RtoL) とのデコーディングアルゴリズムの翻訳結果を示す。日英、英日の翻訳例をそれぞれ図9、図10に示す。

【 0 1 2 3 】

【表2】

翻訳	アルゴリズム	SER[%]	WER[%]	PER[%]	BLEU[%]	NIST	SE [%]			
							A	B	C	D
日英	L to R	14.0	69.7	65.3	65.1	6.85	46.0	22.0	14.0	18.0
	R to L	24.0	70.3	65.0	65.5	6.81	44.0	22.7	16.0	17.3
英日	L to R	13.3	66.3	58.3	65.0	7.63	44.7	12.7	10.7	32.0
	R to L	12.0	66.2	57.6	65.0	7.63	44.0	11.3	10.0	34.7

L to R: left-to-right デコーディングアルゴリズム
R to L: right-to-left デコーディングアルゴリズム

【 0 1 2 4 】

〔 5 〕考察

表2における探索誤り率(SER) から、日英翻訳については、right-to-left デコーディングアルゴリズムより、left-to-right デコーディングアルゴリズムが良く、また、英日翻訳ではその逆の結果が得られた。このことから、言語モデル、翻訳モデルに基づいて翻訳結果を得るに当たって、出力文の言語の特性に合わせて出力の方向を制約する手法の有効性が確認された。

【 0 1 2 5 】

翻訳結果の客観評価値(WER/PER/BLEU/NIST)では、日英翻訳でアルゴリズム間に差異が認められなかった。英日翻訳においては、right-to-left デコーディングアルゴリズムは、WER/PER の点において優れていることが確認された。

【 0 1 2 6 】

主観評価(SE)では、日英翻訳において、left-to-right デコーディングアルゴリズムは、right-to-left デコーディングアルゴリズムよりAランクの文の割合が増加している。ただし、英日翻訳では、アルゴリズム間で主観評価(SE)はほぼ変わらなかった。

【 0 1 2 7 】

【発明の効果】

この発明によれば、文末に語順の制約が強いという性質をもつ言語が目的言語である場合に、好適な翻訳結果が得られるようになる。

【図面の簡単な説明】

【図1】統計的機械翻訳モデルを説明するための模式図である。

【図2】アライメントの例を示す模式図である。

【図3】デコーディングアルゴリズムを示すフローチャートである。

【図4】図3のステップ7の仮説生成処理手順を示すフローチャートである。

【図5】 right-to-left デコーディングによる、openに対応した仮説生成処理を説明するための模式図である。

【図6】 left-to-right デコーディングによる、openに対応した仮説生成処理を説明するための模式図である。

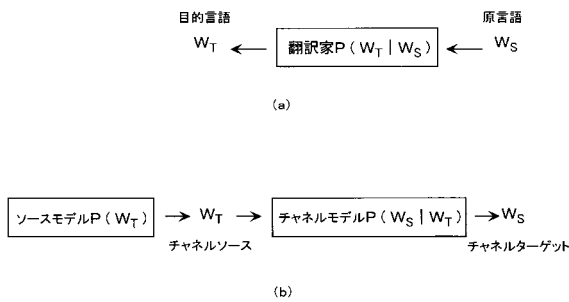
【図7】 right-to-left デコーディングによる、close に対応した仮説生成処理を説明するための模式図である。

【図8】 left-to-right デコーディングによる、close に対応した仮説生成処理を説明するための模式図である。

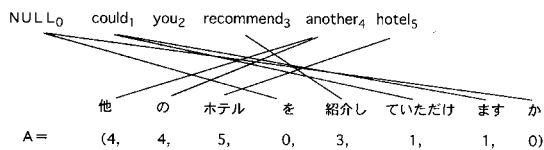
【図9】実験によって得られた日英翻訳結果の例を示す模式図である。

【図10】実験によって得られた英日翻訳結果の例を示す模式図である。

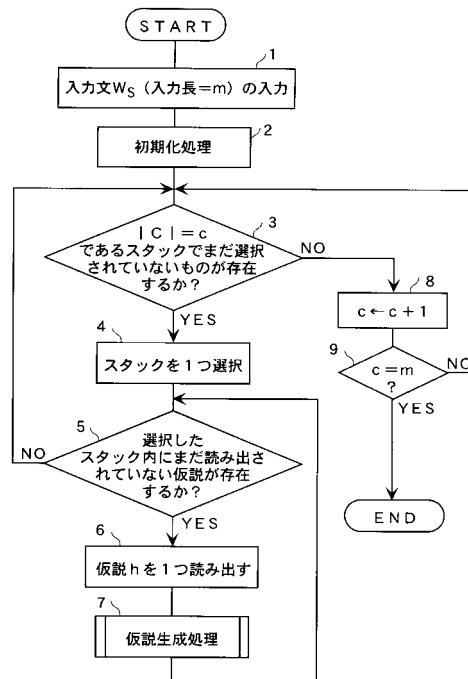
【図1】



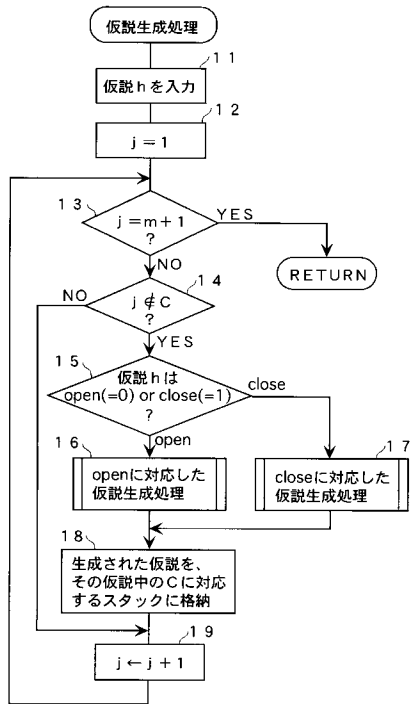
【図2】



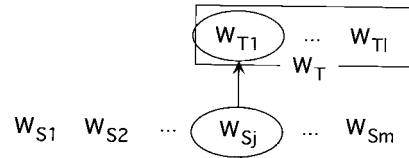
【図3】



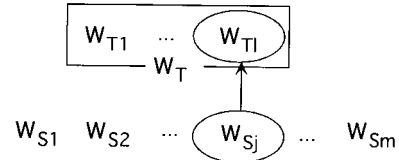
【 図 4 】



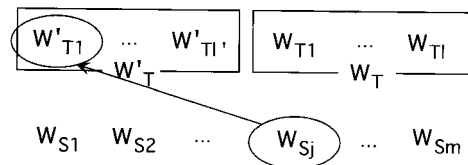
【 図 5 】



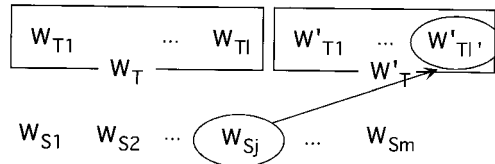
【 図 6 】



【 図 7 】



【 図 8 】



【 図 9 】

入力：	スリに財布をすられました
L to R：	here's my wallet was stolen
R to L：	here's my wallet was stolen
入力：	すみませんがテラスの席がいいのですが
L to R：	excuse me i'd like a seat on the terrace
R to L：	i'd prefer excuse me
入力：	何時に終るのです
L to R：	what time should i be at the end
R to L：	it's what time will it be over
入力：	荷物を上にあげてもらえますか
L to R：	could you put my baggage here
R to L：	do you have overhead luggage
入力：	ええ兄と妹が一人ずついます
L to R：	yes brother and sister there a daughter
R to L：	you're yes brother and sister daughter

【 図 10 】

入力：	this is my boss mr. kitano
L to R：	これは私の上司であります
R to L：	北野上司です
入力：	i'm glad that we have a hobby in common
L to R：	私の趣味が痛いと言ってうれしいです
R to L：	私の趣味熱があつてうれしいです
入力：	excuse me sir what do you want for drink befor dinner
L to R：	すみませんが食事の前に何か飲み物をお願いしますか
R to L：	いくらになります食事の前に飲み物が欲しいのですか
入力：	sorry but i don't understand these medical words in english
L to R：	すみませんこの言葉は英語がわかりません
R to L：	診断書あいにくこの言葉は英語がよくわかりません
Bi	すみませんこの言葉は英語がわかりません
入力：	hurry up or i'll be late for the flight
L to R：	その便はどのくらい遅れますかそれとも急いでください
R to L：	飛行機の急いでいますかそれとも遅れます