

編集距離を使った用例翻訳の高速検索方式と翻訳性能評価

土居 誉生[†] 隅田 英一郎[†] 山本 博史[†]

本稿では編集距離を使った用例翻訳システムの実装方式と大規模コーパスを使った翻訳性能評価について報告する。この用例翻訳方式では、コーパスの中から最も類似した文を選び出す仕組みがその中核にある。その実装のための効率的で検索もれのない手法を提案する。提案手法では検索対象領域の分割と単語グラフ表現、A*アルゴリズムを利用する。翻訳性能評価では旅行会話ドメインの数10万文規模のコーパスを使用する。まず日英翻訳で、3種類の言語資源（対訳コーパス、対訳辞書、シソーラス）と翻訳性能の関係について検証する。また複数の言語対について翻訳システムを構築することにより多言語適用性を確認する。

Efficient Retrieval Method and Performance Evaluation of Example-based Machine Translation Using Edit-distance

TAKAO DOI,[†] EIICHIRO SUMITA[†] and HIROFUMI YAMAMOTO[†]

In this paper, we report on the implementation method of the EBMT using edit-distance and the evaluation of its performance using a large-scale corpus. The core function of the EBMT is to retrieve the most similar sentences from a corpus. An efficient retrieval method is indispensable for implementation. We propose an efficient retrieval method without omissions. The proposed method utilizes division of search space, word graphs and an A* search algorithm. In the experiments for the performance evaluation, we use a corpus of hundreds of thousands of sentences from a travel conversation domain. In Japanese-to-English translation, we investigate correlations between the performance and three linguistic resources: a bilingual corpus, a bilingual dictionary and thesauri. Moreover, we verify the adaptability of the EBMT to multilingual translation by making translation systems for several language pairs.

1. はじめに

アナログに基づく機械翻訳¹³⁾の概念が提唱されて以来、このアイデアを具体化した用例翻訳方式が数多く提案されてきた^{4),9),18),21),27),28)}。用例翻訳では対訳コーパスから対訳表現を抽出し、翻訳すべき文を構成する各部分に適合する対訳表現を見つけ、それらを組み合わせて翻訳文を生成する。用例翻訳以前の代表的な翻訳方式であるルールベース翻訳では、言語現象について人間が内省することにより作成した翻訳のためのルールに基づいて翻訳を実行する。ルールベース翻訳では翻訳ルールの精密な構築が翻訳システムの能力を決定するのに対し、用例翻訳では対訳コーパスからの学習が能力を決める。用例翻訳の大きな利点は、人手による翻訳ルールの記述が不要であり、翻訳システムの保守を含めた開発の効率が高いことである。こ

の特徴により異なるドメインへの移植や新たな言語対への適用が容易になる。

Sumita²³⁾は用例翻訳の1つとしてDPマッチに基づいた翻訳方式(DP-match Driven transDucer, 以下、D³と記す)を提案し、旅行会話における日英翻訳を対象とする評価実験で有効性を示した。本方式では対訳用例は原言語文と目的言語文の対であり、各文は単語列として表現される。翻訳実行時には入力文との類似度の最も高いと判定される原言語文を持つ用例を検索する。検索された用例の原言語文と入力文の差異、差異部分と対応する目的言語文の部分をもとに翻訳パターンを生成する。この際DPマッチが主導的役割を果たす。つまりDPマッチは、文どうしの距離を示す単語列編集距離を与えると同時に、差分箇所を指示し翻訳パターン抽出を可能にする。他の用例翻訳方式と比べたときのD³の特徴は、1)多くの方式ではあらかじめ翻訳パターンを作成しているが、本方式では翻訳実行時に作成する、2)多くの方式では構文解析やツリーバンクの利用を仮定しているが、本方式では利用

[†] ATR 音声言語コミュニケーション研究所
ATR Spoken Language Translation Research Laboratories

しない、という点にある．このように D^3 では、用例を抽象化せず単語列の形のまま保持し検索する．さらに訳文を生成する際にも用例の目的言語文の変更を最低限にとどめるため、入力文に近い用例が存在すれば自然な表現の翻訳結果が得られる．また、日本語や英語以外の多くの言語で高精度パーザが利用できない現状を考えると、本方式は解析知識を使わないため用例翻訳方式の中でも特に多言語適用性に優れる．

本稿では D^3 の実装方式と大規模コーパスを使った評価について報告する． D^3 では、記憶した文の集合の中から最も類似したものを選び出す仕組みがその中核にある．特に大量の用例を利用する場合、その仕組みの効率的な実装が必須となる．そこで実装方式に関しては用例検索処理に焦点を当て説明する．用例検索処理の効率的な実装は翻訳メモリ課題と共通である．翻訳メモリに関する研究^{2),6),16),19)}では、1) 検索対象の絞り込み、2) 2文間の照合アルゴリズム、の一方もしくは両方を論じている．1)を行う場合、単語ベクタを使ったクラスタリングなどの手法により文を絞り込み、候補として残った文のそれぞれと入力文との2文間の照合を繰り返す．絞り込みの際検索もれが生じる．また翻訳メモリに関する他の研究¹⁷⁾では品詞レベルでの文の完全一致に基づいた検索を提案している．この方法では単語の挿入、削除や意味の類似度を考慮した柔軟な照合に対応できない．本稿では、用例の集合の中から単語列編集距離最小の用例を探索する問題の解法として、効率的で検索もれのない手法を提案する．この手法では2文間の逐次的な照合は行わず、グラフ化された複数の候補文と入力文との照合を同時並行的に進める．

評価実験では、旅行会話に関する数10万文規模のコーパスを用い、言語資源と翻訳性能の関係、多言語適用性について検証と議論を行う． D^3 は対訳コーパス、対訳辞書、シソーラスを利用する．用例翻訳方式による翻訳システム開発のコストは、必要となる言語資源を準備するためのコストに依存する．ここでは言語資源の量と作成のための人手の有無と翻訳品質の関係を考察する．また大量の用例を用いる場合、用例検索時間の増大が懸念される．大規模コーパス使用時に実用的な処理時間で翻訳可能か検証する．また複数の言語対を用い多言語適用性を確認する．

以下、2章で D^3 の概要、3章で用例検索の概要、4章でその探索アルゴリズム、5章で翻訳実験とその評価について述べる．

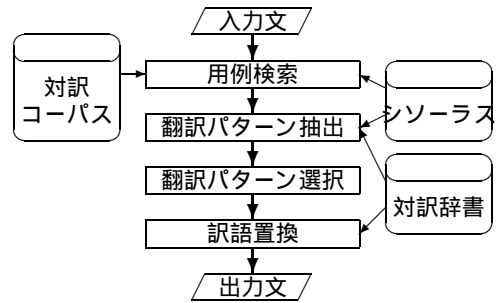


図1 D^3 の構成
Fig. 1 Configuration of D^3 .

2. D^3 の概要

2.1 構成

図1に示すように、 D^3 は対訳コーパス、対訳辞書、シソーラスの3種類の言語資源を使用する．対訳コーパスは、翻訳対象の原言語と目的言語の文の対の集合である．両言語の文は互に対訳関係にある．対訳コーパス中の文は単語に分割され品詞情報が付与されている． D^3 では、対訳コーパス中の対訳関係にある文のペアを利用して翻訳を実行する．以下、この文のペアを用例と呼ぶ．対訳辞書は翻訳パターン抽出と訳語置換処理で使われる．シソーラスは、原言語と目的言語それぞれに用意され、用例検索および翻訳パターン抽出処理で使われる．

2.2 アルゴリズム

同じく図1に示すように、翻訳処理は、(1)用例検索、(2)翻訳パターン抽出、(3)翻訳パターン選択、(4)訳語置換、の4フェーズからなる．以下、各フェーズについて説明する．

2.2.1 用例検索

用例検索処理は全用例の原言語文を走査する．入力文と用例原言語文の単語列間の距離を測り、最小距離の用例を選び出す．この最小距離が大きければ検索された用例は翻訳処理に有用ではない．そのため距離に閾値を設ける．閾値以内の距離の用例が存在しなければ用例検索および翻訳処理は失敗に終わる．単語列間の距離には意味距離の加味された単語列編集距離を使う．この単語列編集距離 $dist$ は次の式で表される．

$$dist = \frac{I + D + 2 \sum SEMDIST}{L_{input} + L_{example}}$$

ここで、 L_{input} は入力文の単語数、 $L_{example}$ は用例原言語文の単語数、 I は挿入単語数、 D は削除単語数、 $SEMDIST$ は置換語の意味距離を示す．この式に従って、挿入語と削除語の数、置換語の意味距離が

足し合わされ、入力文と用例原言語文の長さの和をもって正規化される。置換については、2 単語が同品詞の内容語である場合のみ対象となり、2 単語間の意味距離が単語列編集距離の計算に使われる。意味距離計算には文献 21) の方法を使う。この方法では 2 つの単語に関してシソーラスの概念階層における意味概念間の位置関係によって計算される。意味距離は、0 から 1 までを値域とし、0 に近いほど 2 単語は意味的に類似していることを示す。

次に日英翻訳での例を示す。(1-j) は入力文、(2-j) は用例の原言語文とする。網かけ部分が両文の差分となる。

(1-j) 色 / が / 気 / に / 入り / ません

(2-j) デザイン / が / 気 / に / 入り / ません

ここで「色」と「デザイン」がシソーラス上で完全に異なった語とすると単語間の意味距離は 1 となり、2 文間の単語列編集距離は、 $(0+0+2*1)/(6+6)=0.167$ となる。この 2 文間の単語列編集距離は DP マッチと呼ばれる動的プログラミングの手法により計算可能である⁵⁾。

2.2.2 翻訳パターン抽出

検索された用例の原言語文中の入力文と異なる箇所を変数とし、用例目的言語文中の対応する箇所に同じ変数をあてはめた翻訳パターンを生成する。両言語の文の間で対応をとる際は変数となる単語のみ対象とし、すべての単語の対応をとる必要はない。つまり変数部分以外の箇所は全体として対応していると仮定する。このため用例のほとんどの部分は変更されず、訳文の組合せ時に発生する誤りや不自然さの回避が期待される。この原言語と目的言語の単語間の対応をとるには様々な単語アライメントの手法¹¹⁾の適用が考えられる。今回の実装では対訳辞書、両言語のシソーラスに基づいて単語間の対応関係を判断している。

先の例の (2-j) に対応する目的言語文を (2-e) とする。このフェーズでは (2-e) 中で「デザイン」に対応する箇所が探しだされ、文中の網かけ部分で示された対応がとられる。

(2-j) デザイン / が / 気 / に / 入り / ません

(2-e) I do not like the design.

この結果、原言語パターン (2-j-p) と目的言語パターン (2-e-p) からなる翻訳パターンが作られ、入力文による変数束縛が (1-j-b) となる。

(2-j-p) X / が / 気 / に / 入り / ません

(2-e-p) I do not like the X.

(1-j-b) X = 「色」

2.2.3 翻訳パターン選択

最小距離の用例が複数あり複数種類の翻訳パターンが作られる場合がある。その際、翻訳パターンを選択するために、1) 入力文とパターンの不整合の小さい方を選ぶ、2) より多くの用例検索結果から同じ翻訳パターンが抽出された方を選ぶ、3) 翻訳パターン中に現れる単語のコーパスでの出現頻度の合計が大きい方を選ぶ、というヒューリスティクスを使用する。1) の不整合は、目的言語文と対応のとれない変数の個数、挿入語数、削除語数が大きいほど大きくする。これらで決定できない場合は任意の 1 つの翻訳パターンを選ぶ。

2.2.4 訳語置換

翻訳パターンの変数に束縛された単語の訳語を対訳辞書から引き、その訳語でもって目的言語パターンの変数を具体化する。

先の例を使うと目的言語側の変数束縛は (1-e-b) となり、訳文 (1-e) が得られる。

(1-e-b) X = "color"

(1-e) I do not like the color.

2.2.5 挿入と削除

ここまでのアルゴリズム説明に用いた例では、用例の原言語文と入力文とで異なる部分どうしの対応がとれているが、本方式では対応のとれないパターンも許している。用例側に対応する語を持たない入力文側の語、すなわち挿入語は、翻訳パターンに組み込まれず訳語置換フェーズでは無視される。入力文側に対応する語を持たない用例側の語、すなわち削除語は、束縛されない変数となる。目的言語パターン中に同じ変数があれば、その変数欄には訳語置換処理によって空文字列が入れられ、結果として用例目的言語文中にあった語が翻訳文から削除されることになる。目的言語パターン中に同じ変数がなければ、削除語は生成される翻訳文に影響を与えない。

挿入や削除により入力文や用例の情報の欠落が起こりうるが、単語列編集距離の定義により、意味的に類似した語の置換が挿入や削除よりも優先される。そのうえで挿入や削除を許すことにより、特に話し言葉における助詞の脱落などの現象に対する頑健性を与える。

たとえば用例原文「コーヒー / か / 紅茶 / は / いか / が / です / か」に対し、入力文「ビール / か / ワイン / は / いか / が / です / か」では 2 カ所の置換があるが、両置換において対応する語は飲み物としてシソーラス

今回の実装では訳語選択はしていない。1 つのエントリに対する訳はただ 1 つである。

上一致し、意味距離さらには単語列編集距離は 0 に近い値となる。また同じ用例原文に対して「コーヒー／か／紅茶／いかが／です／か」という入力文が与えられると、係助詞「は」1 語の削除が発生し単語列編集距離は $1/(6+7)=0.077$ となる。より近い用例が見つからなければ、この用例が訳文生成に利用される。逆に用例側の助詞が省略されている場合は助詞の挿入が発生することになる。

3. 用例検索

前章で説明した D^3 の各処理の中で翻訳実行時間の大きな割合を占めるのは用例検索である。用例の選択基準には 2.2.1 項で定義された単語列編集距離が使われる。用例検索処理は、用例の原言語文を候補文とし、入力文との距離が閾値以内で最小の候補文をすべて求めることである。この単語列編集距離は 2 文間の関係で定義されていて、2 つの単語列の DP マッチにより計算可能である。したがって、各候補文と入力文間の DP マッチを逐次的に繰り返すことで最小距離の候補文を求めることができる。しかし単純にこの方法を使うとすれば用例数に比例した処理時間がかかってしまい、大規模用例を利用したリアルタイムの翻訳処理を実現することは通常の計算機では難しい。そこで我々は、候補文集合の分割、単語グラフ、A*アルゴリズムを利用した効率的な実装を提案する。この実装方法では単語列編集距離の定義と与えられた閾値に関して検索もれない。つまり DP マッチを全候補文に対して逐次的に繰り返した場合と同じ距離最小の候補文の集合を検索結果として返す。

3.1 候補文集合の分割

内容語数と機能語数を基に候補文をグループ分けする。これにより入力文の内容語数と機能語数および距離閾値により検索対象の候補文数を絞ることができる。つまり、機能語どうし、内容語どうしはすべて一致すると仮定した場合のグループごとに可能な最小距離を求める。最小距離が距離閾値の範囲内で小さいグループから順に検索する。あるグループ中に解が見つければ、その解の距離を新たな閾値として検索対象のグループはさらに絞られる。また、グループ内での処理については 4 章で説明するが、グループ内ではすべての候補文の内容語数と機能語数が等しい、つまりは単語数も等しいということを前提条件として利用する。

3.2 単語グラフ

内容語数と機能語数を基準に分けられたグループごとに複数の候補文が 1 つの単語グラフにまとめられる。図 2 に単語グラフの例を示す。単語グラフは有

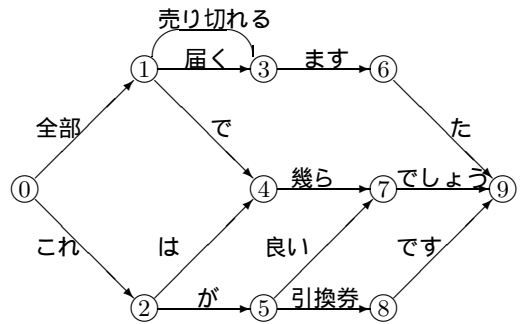


図 2 単語グラフの例
Fig. 2 Example of word graph.

向グラフであり、先頭ノードから最終ノードに至る可能な道筋と候補文が互に対応する。複数の文に共通な単語列がグラフ中で 1 つにまとめられ、ノード数が最小となるように圧縮されている。この圧縮処理には有限状態オートマトンを変換する手法³⁾が使われている。単語グラフを利用することにより、グループ内の全候補文を同時並行的に調べながら、入力文との距離が最小の候補文を検索する。

3.3 A*アルゴリズム

2 つの単語列を照合した結果を示す単語の一致、置換、挿入、削除の列を照合列と呼ぶこととする。グループ内の検索は、単語グラフの先頭ノードから最終ノードまでの可能な全経路について、各経路に現れる単語列と入力単語列との照合列の中から単語列編集距離を最小にするものを探索することである。この探索問題の解法に A*アルゴリズムを用いる。一般に A*アルゴリズムでは、問題状態集合の中から最終コストの下限の推定値が最小のものが選ばれ継続状態に展開される。ここで対象とする問題では、状態は、単語グラフの経路と入力文との照合の途中経過を意味する。

4. 探 索

本章では 1 つの単語グラフを使った探索処理に絞って説明する。単語グラフはノードとリンクからなる。リンクは単語をラベルとして持ち、1 つの始点ノードと 1 つの終点ノードを結び、単語グラフ全体で 1 つの先頭ノードと 1 つの最終ノードを持つ。

4.1 状態空間表現

対象となる問題の状態空間は以下で説明する状態、作用素、初期状態と目標状態により構成される。

4.1.1 状 態

状態は paths, node, input, trans の属性を持つ。各属性の内容は以下のとおりである。

- paths: その時点までの照合列のリスト。

- node: 単語グラフのノード. このノードまで照合が進んだことを示す.
- input: 入力単語列のいまだ照合に使われていない部分.
- trans: 適用可能な作用素を示す.

paths 内の各照合列の一致, 置換, 挿入, 削除をそれぞれ, (E, 単語), (S, グラフ側単語, 入力側単語), (I, 入力側単語), (D, グラフ側単語) の形式で表し, それぞれ Eレコード, Sレコード, Iレコード, Dレコードと呼ぶ. 状態のコストは paths 内の任意の 1 つの照合列のコストである. paths 中のどの照合列も等しいコストを持つ. 照合列のコストは, それに含まれるレコードのコストの和であり, 各レコードのコストは Eレコードは 0, Iレコードは 1, Dレコードは 1 と定義される. Sレコードのコストは 2 単語間の意味距離を 2 倍した値であるが, 意味距離が 0 の場合には小さな正の値を与える. この値が Sレコードの最小コストとなる.

4.1.2 作用素

状態に作用素を適用することにより継続状態が生成される. 一般に 1 つの状態に複数の作用素が適用可能であり, 1 つの状態からいくつかの継続状態が生成される. 下で 5 種類の作用素を定義する. T 作用素と I 作用素は状態に適用されるが, E, S, D の各作用素は状態, および, その状態の node を始点とするリンクの組に適用される. 以下の説明では, 作用素の適用される状態を s , リンクを l , 生成される継続状態を s' と表し, 各作用素について適用条件とどのような継続状態が生成されるかを記述している.

- T 作用素:
 - 条件: $s.trans$ が E 作用素または S 作用素である.
 - 生成: $s'.trans = s.trans$ が E 作用素ならば S 作用素と NIL から選択(説明は後述), $s.trans$ が S 作用素ならば NIL
 s' の他の属性値は s と同じ
- E 作用素:
 - 条件: $s.trans$ が E 作用素である.
かつ, $s.input$ が空リストでない.
かつ, l のラベルと $s.input$ の先頭が同一語である.
 - 生成: $s'.paths = s.paths$ の各要素に Eレコードを追加した値
 $s'.node = l$ の終点

$s'.input = s.input$ から先頭を消去した値
 $s'.trans = E$ 作用素と S 作用素と NIL から選択(説明は後述)

- S 作用素:
 - 条件: $s.trans$ が S 作用素である.
かつ, $s.input$ が空リストでない.
かつ, $s.input$ の先頭と l のラベルが同品詞の内容語であり, かつ, 同一語ではない.
かつ, これら 2 単語の意味距離は 1 未満である.
 - 生成: $s'.paths = s.paths$ の各要素に Sレコードを追加した値
 $s'.node = l$ の終点
 $s'.input = s.input$ から先頭を消去した値
 $s'.trans = E$ 作用素と S 作用素と NIL から選択
- I 作用素:
 - 条件: $s.trans$ が NIL である.
かつ, $s.input$ が空リストでない.
 - 生成: $s'.paths = s.paths$ の各要素に Iレコードを追加した値
 $s'.node = s.node$
 $s'.input = s.input$ から先頭を消去した値
 $s'.trans = E$ 作用素と S 作用素と NIL から選択
- D 作用素:
 - 条件: $s.trans$ が NIL である.
かつ, $s.paths$ に最新レコードが Iレコードでない要素がある.
 - 生成: $s'.paths = s.paths$ から最新レコードが Iレコードである要素を除き, 残った要素に Dレコードを追加した値
 $s'.node = l$ の終点
 $s'.input = s.input$
 $s'.trans = E$ 作用素と S 作用素と NIL から選択

上記で「S 作用素と NIL から選択」とは, s' に S 作用素を適用できる可能性があれば $s'.trans$ の値を S 作用素とし, 可能性がなければ NIL とすることである. ここでは, $s'.input$ の先頭が内容語であり, その語との同一語を除く同品詞語をラベルとし $s'.node$ を始点とするリンクが存在する場合に S 作用素を適用できる可能性がある判断する. また「E 作用素と S 作用素と NIL から選択」とは, $s'.input$ の先頭語をラベルとし $s'.node$ を始点とするリンクが存在すれば $s'.trans$ の値を E 作用素とし, そうでなければ S 作用

素と NIL から選択する．T 作用素は実際に照合を進める作用素ではなく，trans 属性とともに E, S, I, D の各作用素の適用順序を制御する役目を持つ．D 作用素の 2 番目の条件は I レコードの後に D レコードが来るのを防いでいる．つまり，I レコードと D レコードが連続する場合 D が先に来るようにし，実質的に同じ照合列が複数現れる冗長性を排除する．

4.1.3 初期状態と目標状態

初期状態では paths は空リストを要素とするリスト，node は先頭ノード，input は入力単語列全体，trans は E 作用素となる．目標状態は，node が最終ノード，かつ，input が空リストであるような状態である．

4.2 探索アルゴリズム

上記の初期状態，作用素，目標状態で表現される状態空間からコスト最小の目標状態を探索する．初期条件としてコスト上限値が与えられる．コスト上限値は入力文長と候補文長の和を距離閾値に乗じた値である．

4.2.1 評価関数

状態空間探索時に使用する評価関数 f^* を次のように定義する．

$$f^*(s) = g(s) + h^*(s)$$

$g(s)$ は初期状態から状態 s に達するまでにかかったコスト，つまり先に定義した状態のコストであり $s.paths$ から計算できる．目標状態では $f^*(s) = g(s)$ となる． $h^*(s)$ は状態 s から目標状態までにかかるコストの下限である．1 つの単語グラフを構成する全候補文の内容語数，機能語数はそれぞれ同一であるため，状態 s において入力文側とグラフ側の未処理の内容語数，機能語数は一意に決まる．それぞれの個数を C_{input} ， C_{graph} ， F_{input} ， F_{graph} ，として，残り語数に基づく最小コスト $h'(s)$ を次のように計算する．

$$h'(s) = |C_{input} - C_{graph}| + |F_{input} - F_{graph}|$$

さらに，T 作用素の適用が先行する場合を含めて，状態 s に最初に適用可能な E, S, I, D の各作用素について，それが適用されたと仮定したときの目標状態までにかかるコストの下限を次の値とする．

- E 作用素： $h'(s)$
- S 作用素： $h'(s)$ に S レコードの最小コストを加えた値．
- I 作用素： $s.input$ の先頭が内容語の場合は， $|C_{input} - 1| - C_{graph}| + |F_{input} - F_{graph}|$ に 1 を加えた値，機能語の場合は， $|C_{input} - C_{graph}| + |(F_{input} - 1) - F_{graph}|$ に 1 を加えた値．
- D 作用素： $|C_{input} - (C_{graph} - 1)| + |F_{input} - F_{graph}|$ と $|C_{input} - C_{graph}| + |F_{input} - (F_{graph} - 1)|$ を求

め，その小さい方の値に 1 を加えた値．ただし $s.node$ を始点とするリンクのラベルが内容語のみであるか機能語のみであれば，対応する一方の値に 1 を加えた値．

これらを使って $h^*(s)$ を，1) $s.trans$ が E 作用素のときは，E 作用素が適用されたときのコストの下限，2) $s.trans$ が S 作用素のときは，S 作用素，I 作用素または D 作用素が適用されたときのコストの下限の最小値，3) $s.trans$ が NIL のときは，I 作用素または D 作用素が適用されたときのコストの下限の最小値，とする．

4.2.2 アルゴリズム

探索アルゴリズムは下のように表される．この記述中，OPEN は未展開状態を，CLOSED は展開済み状態を保持するためのリストを示す．また (5) における「同じ状態」とは，paths を除く属性値の等しい状態を意味する．

- (1) コスト上限を与えられた値にする．OPEN に初期状態のみを入れる．
- (2) OPEN にコスト上限以下の状態がなければ終了．
- (3) OPEN から評価関数 f^* を最小にする状態 s を取り除き，CLOSED に入れる．
- (4) s が目標状態なら，それを解の 1 つとし，コスト上限を s のコストに変え，(2) に戻る．
- (5) s のすべての継続状態を生成する．各継続状態 s' について， $f^*(s')$ がコスト上限以下であれば，OPEN および CLOSED 中の同じ状態と比較し，
 - (a) 同じ状態がなければ， s' を OPEN に追加．
 - (b) s' よりコストの大きい同じ状態が OPEN または CLOSED に既存であれば，この既存状態を消去し， s' を OPEN に追加．
 - (c) s' とコストの等しい同じ状態が CLOSED に既存であれば，この既存状態を消去し， s' を OPEN に追加．
 - (d) コストの等しい同じ状態が OPEN に既存であれば，この既存状態の paths に $s'.paths$ をマージする．
- (6) (2) に戻る．

4.2.3 単語グラフの特徴の利用

単語グラフの形状の特徴として，先頭ノードを始点とするリンク数は他のノードを始点とするリンク数よりも圧倒的に大きくなる傾向がある．そのため node 属性に先頭ノードを持つ状態に D 作用素が適用されると多くの継続状態が作られることとなり計算時間が

大きくなる．これは照合列の先頭要素が D レコードとなる場合である．この展開数の増大を避けるため，単語グラフ中，先頭ノードから数段階の仮のリンクとノードを加える．先頭ノードを持つ状態から D 作用素によって第 1 の仮のノードを持つ状態へ遷移する．第 1 の仮のノードは，全候補文について 2 番目の語をラベルとするリンクの始点となり，通常の単語グラフのノードに合流する．第 1 の仮のノードにある状態は E 作用素または S 作用素の適用により通常のノードの状態に，D 作用素によって第 2 の仮のノードを持つ状態に遷移する．

何段階まで仮のノードを用意するかは，用例検索時に使われる可能性のある距離閾値の最大値から計算できる．候補文の長さを L とすると，照合列の先頭に D レコードが d 個並ぶという条件で，候補文との距離を最小にする入力文は，候補文から先頭 d 語を除いた文である．そのときの距離は $d/((L-d)+L)$ である．この値が距離閾値の最大値を超える場合は探索する必要がない．距離閾値の最大値を θ とすると $d/((L-d)+L) \leq \theta$ から $d \leq 2\theta L/(1+\theta)$ が導かれる．この式を満たす d の最大の整数値が用意すべき仮のノードの段数である．

4.3 実行例

探索の実行例を示す．図 2 の単語グラフから入力文「全部揃いました」の類似文を検索することにする．ここでは状態を [paths, node, input, trans, f^* 関数値] の形式で記述する．node 値には図 2 中でノードに付けた番号を用いる． δ を S レコードの最小コストとする．また「揃う」と「売り切れる」の意味距離を 1.0，「揃う」と「届く」の意味距離を 0.7 であると仮定する．初期状態 s_0 は次のようになる．

$s_0 = [((), ノード 0, (全部, 揃う, ます, た), E 作用素, 0]$

s_0 に適用可能な作用素は E 作用素と T 作用素である．これらの作用素を適用して継続状態 s_1 と s_2 が得られ，OPEN は $\{s_1, s_2\}$ となる．ノード 0 を始点とするリンクのラベルには同一語を除いて「全部」の同品詞語がないため $s_2.trans$ は NIL になる．

$s_1 = [(((E, 全部))), ノード 1, (揃う, ます, た), S 作用素, \delta]$

$s_2 = [((), ノード 0, (全部, 揃う, ます, た), NIL, 2]$

ここで OPEN から f^* 関数値の小さな s_1 が選ばれ展開される． s_1 に適用可能な作用素は S 作用素と T 作用素である．「売り切れる」と「届く」をラベルとする 2 つのリンクに関して S 作用素の適用条件がテ

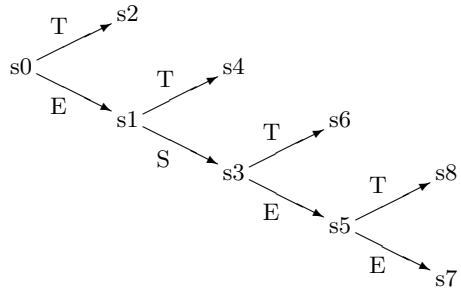


図 3 状態遷移図
Fig. 3 State chart.

表 1 コーパスサイズ
Table 1 Statistics of the corpus.

	日本語	英語	韓国語	中国語
文数	404,022	404,022	172,468	172,468
総語数	2,870,280	2,473,711	871,038	956,578
語彙数	33,288	22,378	19,599	15,367

ストされるが，1 未満の意味距離の得られる「届く」のリンクについてのみ成功する．S 作用素の適用により継続状態 s_3 が得られる．また T 作用素の適用により継続状態 s_4 が生成される．OPEN は $\{s_2, s_3, s_4\}$ となる．

$s_3 = [(((E, 全部), (S, 届く, 揃う))), ノード 3, (ます, た), E 作用素, 1.4]$

$s_4 = [(((E, 全部))), ノード 1, (揃う, ます, た), NIL, 2]$

ここで OPEN から f^* 関数値が最小の s_3 が選ばれ展開される． s_3 に E 作用素を 2 回適用した s_7 が解となる．

$s_7 = [(((E, 全部), (S, 届く, 揃う), (E, ます), (E, た))), ノード 9, (), NIL, 1.4]$

この例の状態遷移の様子を図 3 に示す．T 作用素の適用による遷移を除き解に向かって一直線に探索が進んでいる．

5. 翻訳実験

大規模コーパスを使った翻訳実験により D^3 の性能を検証した．検証項目は，1) 基本的な翻訳性能，2) 言語資源と翻訳性能との関係，3) 多言語適用性，である．なお 1), 2) には日英翻訳，3) には，日韓，日中，英日，韓日，中日の各方向の翻訳を用いた．

5.1 実験条件

本実験で用いたコーパスは，ATR の旅行会話基本表現集²⁵⁾である．このコーパスは，旅行会話に頻出する表現を集めたものである．このコーパスのサイズを表 1 に示す．

表 2 訓練セットとテストセット
Table 2 Training set and test set.

	日本語	英語	韓国語	中国語
訓練セット (異なり)	97,116	91,842	97,908	86,864
テストセット (異なり)	502	503	498	497
一致テスト文数 (のべ)	227	210	202	260

表 3 日英翻訳性能
Table 3 JE translation performance.

mWER	BLEU	主観評価ランク (%)			出力率 (%)	処理時間 (秒)	
		A	AB	ABC		平均	最大
0.3425	0.6399	71.2	80.4	83.3	91.8	0.218	3.310

このコーパスでは、日本語、英語、韓国語、中国語のそれぞれが文単位で対応している。ただし日英の文数が韓中より多く、対応する韓国語と中国語を持たない日英だけの文からなる部分もある。つまり日英韓中の4カ国語の文が172,468組、日英のみの文が231,554組ある。実験では日英韓中の文の組から152,170組の訓練セットと510組のテストセットを切り出し使用する。ただし、5.3.1項の実験で訓練セットのサイズを拡張する場合、日英のみの文からなる組も訓練セットに追加して使用する。4カ国語の文は人手によって作成され、それぞれの言語の形態素解析プログラムを使って単語に分割され品詞情報が付与されている。各言語対の翻訳実験では、訓練セット中の原言語文と目的言語文を抜き出し、対訳コーパスとしてD³の用例を提供する。またテストセット中の原言語文が翻訳すべき入力文となる。

コーパスでは同じ文が重複して現れる場合がある。文の出現頻度はその文がドメイン内の会話で使われる可能性の高さを示すと考えられ、実験では出現頻度の違いを考慮する。つまり訓練セットとテストセットの抽出に際しては、重複を含んだ全コーパスの中からランダムに文の組を抜き出し、重複の除去は行っていない。翻訳パターン選択処理では文の重複を含めた対訳用例や単語の頻度情報を利用する。原言語文が同じで目的言語文が異なる複数の対訳の中からどれを使って訳文を生成するかは翻訳パターン選択処理で決められる。表2は訓練セットとテストセットそれぞれに含まれる異なり文数を示す。表中の一致テスト文数は、テストセットの文で訓練セットに同一文の存在する文の数を、テストセット510文に関するのべ数で表している。

日英・英日および日韓・韓日の対訳辞書には、他の翻訳システム²²⁾用に準備された旅行会話ドメインの既存辞書を使用した。日中・中日についても、他の翻

訳システム²⁹⁾用に準備された旅行会話ドメインの既存辞書を使用した。日本語と英語のシソーラスには、同じく他の翻訳システム²²⁾用に準備された角川類語新辞典¹⁴⁾のシソーラス構造に準拠したものを使った。韓国語と中国語のシソーラスは使わなかった。

翻訳結果の品質評価のためには、客観評価スコアと主観評価ランクを用いた。客観評価スコアには多重参照単語誤り率²⁶⁾(multi-reference Word Error Rate, 以下、mWERと記す)とBLEUスコア¹⁵⁾を使用した。mWERは翻訳結果が参照訳と異なる割合を編集距離に基づいて計算する。BLEUは翻訳結果と参照訳のNグラム的一致する割合を示す。翻訳結果の品質が高くなればmWERは小さくなり、BLEUスコアは大きくなると考えられている。mWER、BLEUともに各テスト文につき16文の正解翻訳例を参照訳とした。主観評価では、原言語を理解できる目的言語のネイティブにより4段階の評価を与えた²²⁾。評価ランクは、(A)問題なし、(B)主要な情報が容易に復元できる、(C)主要な情報がかるうじて復元できる、(D)主要な情報が復元できない、としている。

各実験でのD³の距離閾値は特に触れない限り1/3とした。また実験にはPentium4/2GHzのコンピュータを使い、プログラムはAllegro Common Lisp 6.2版上で実装、実行した。

5.2 基本的な翻訳性能

日英翻訳の実験結果を表3に示す。表中、主観評価欄は良い方のランクからの累計数の全テスト文数に対する割合を示している。出力率とABCの差はDランクとなる。Aランクの割合が大きく品質の高い翻訳文が得られていることが分かる。一方で出力率は91.8%であり約8%の文で翻訳結果が出力されていない。

翻訳処理時間は、平均0.2秒、最大3.3秒であり、提案した用例検索手法により効率的な処理が実現されて

表 4 D³ と人間の対比較Table 4 Translation accuracy: D³ vs. human.

TOEIC スコア	D ³ の勝ち	人間の勝ち	引き分け
420	224	133	153
540	214	149	147
685	211	150	149
820	176	157	177
965	141	173	196

いる．なお用例検索処理のために作られた単語グラフの個数は 335 であり，1 つの単語グラフ内の平均ノード数は 728，平均リンク数は 996 であった．

5.2.1 人間の英語能力との比較

この翻訳品質を人間と比較するとどのレベルに相当するか，どのレベルの英語能力を持った日本人が同等の翻訳文を作れるのかを文献 20) の方法で求めた．この方法では，いろいろなレベルの TOEIC スコア (Test Of English for International Communication) を持つ人間の作った翻訳文とシステムの翻訳出力を一对比較法で評価し，回帰分析により人間とシステムの能力の均衡する TOEIC スコアを計算する．

今回は TOEIC スコア 420 点から 965 点までの 5 人の日本人と比較した．表 4 に一对比較の結果を示す．ここから D³ の能力として 870 点の TOEIC スコアが得られた．日本企業で海外と取引のある部門の社員の平均的なスコアが 750 点，860 点以上で非英語ネイティブとしては最高レベルと能力評価されていることから見ると，この実験では D³ は高い品質を示したといえる．

5.2.2 編集距離と翻訳品質

図 4 は単語列編集距離と主観評価ランクとの関係を示す．用例と入力文との距離が大きくなるに従って，A ランクの割合は小さくなり，逆に D ランクの割合は増加する．距離と翻訳品質との間に明瞭な相関があることが分かる．

5.2.3 翻訳誤りの分析

表 5 に翻訳の成功例，表 6 に典型的な失敗例をいくつか示す．ここでは失敗例に注目する．

(1-a)，(1-b)，(1-c) では入力文に含まれない情報が翻訳結果に現れる現象が起こっている．(1-a) の翻訳に使われた用例は (メキシコに行くつもりです → I'm going to visit Mexico after leaving the U.S.) であり，この対訳に限られた場面でしか使えない特殊な対訳であることが原因で翻訳が失敗している．(1-b) の用例は (便名と時間を教えて下さい → What is the flight number and departure time?) であり，原言語と目的言語間の単語の対応付けに失敗している．すなわち

「時間」と“time”が対応すると判断され“departure”が残ってしまっている．(1-c) の用例は (無農薬のオレンジはありますか → Do you have pesticide free oranges?) である．これも単語対応付けの失敗である．実験で使用している対訳辞書では「無農薬」と“pesticide free”との対応がとれず，シソーラスを使って「無農薬」と“pesticide”の対応がとられ“free”が残ってしまっている．なお，この例では原言語側で「赤」と「オレンジ」は色の意味で類似語と判断され対応がとられている．これらの問題への対策として，極度に文脈依存した用例や誤った用例を排除するために，原言語と目的言語で逐語的な対応のとれない対訳を判定する手法¹⁰⁾が有望である．また 2.2.2 項で述べたように，現状では原言語と目的言語の単語間の対応をとるために対訳辞書とシソーラスに基づく手法を使っているが，より良いアライメントの手法による改善が期待できる．

(2-a)，(2-b)，(2-c) では入力文の情報が翻訳結果から欠落している．(2-a) の用例原文は「聞きもらしたのですが」であり，入力文の「お名前を」が無視されている．(2-b) の用例原文は「この小包を日本に送りたいのですが」であり入力文の後半が無視されている．(2-c) の用例原文は「私はジャズが好きです」であり入力文の「ハウスや」が無視されている．これらは用例原文に対する挿入語の問題である．現状ではすべての挿入語に距離計算上同じコストを与えているが，語の重要度を測る何らかの指標を導入しコストに差を付けることにより，重要な情報を落ちにくくする対策が考えられる．挿入部が長い場合には入力文を分割する手法⁸⁾も有望である．また (2-c) で「ハウスやソウルミュージック」のような句全体を「ジャズ」と対応させることも考えられるが，そのためには句を扱う仕組みと協調させる必要がある．

(3) は訳語の誤りである．対訳辞書に従って「名」の訳語が“name”になっているが，この場合不適切である．これは現在の実装では訳語選択を行っていないためであり，単語の共起関係を基にした手法など何らかの訳語選択の導入によって改善できると期待される．

なお主観評価が D ランクとなった 43 文において，(1-a)，(1-b)，(1-c) のタイプの誤りが 26 文，(2-a)，(2-b)，(2-c) が 15 文，(3) が 8 文で観察される．同一文で複数のタイプの誤りが現れる場合もある．また誤りがこれら 3 つのタイプのいずれでもない文が 4 文ある．この 4 文を除く 39 文について，上述の各対策による改善の可能性はある．

もちろんいずれの問題も適切な用例の追加による改

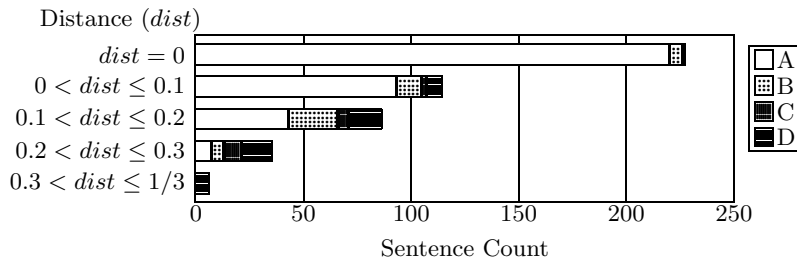


図 4 単語列編集距離と翻訳品質
Fig. 4 Edit-distance and translation accuracy.

表 5 翻訳の具体例
Table 5 Sample translations.

入力文 → 翻訳結果 (距離)
これの色違いありませんか。 → Do you have this in different colors? (0.077)
直し代九十九ドルいただきます。 → There is a ninety nine dollar repair charge. (0.111)
今朝朝食以外に何かサービスをお受けになりましたか。 → Did you have any other services besides breakfast this morning? (0.200)
仕事それとも観光。 → Are you on business or pleasure? (0.250)

表 6 翻訳誤りの典型例
Table 6 Typical incorrect translations.

(1-a) ボストン美術館に行くつもりです。 → I'm going to visit Boston Museum <u>after leaving the U.S.</u>
(1-b) お名前とお部屋番号を教えてください。 → What is the name and <u>departure</u> room number?
(1-c) ワインの赤はありますか。 → Do you have wine <u>free</u> red?
(2-a) お名前を聞きもらしたのですが。 → I <u>m</u> issed that?
(2-b) この小包を日本に送りたいのですが一番はやい方法はありますか。 → I'd like to send this parcel to Japan.
(2-c) 私はハウスやソウルミュージックが好きです。 → I like soul music.
(3) 二名です禁煙席をお願いします。 → Just two <u>n</u> ame, nonsmoking section please.

善の可能性はある。5.3.1 項の実験で訓練セットを拡張したとき (2-b) の翻訳結果は “What is the quickest way to send this package to Japan?” に改善されている。

5.3 言語資源と翻訳性能

D³ の使用する 3 種類の言語資源である対訳コーパス、対訳辞書、シソーラスに関して条件を変えて実験を行った。各実験結果では主観評価ランクと mWER はよく相関しているが、BLEU スコアによる評価は他の評価基準と大きく食い違う場合がある。ここでは翻訳品質に関する議論は主観評価ランクに基づいて行うことにする。

5.3.1 対訳コーパス

訓練セットのサイズを 2 倍、1/2、1/4、1/8 としたときの翻訳結果の品質および翻訳処理速度を評価する。表 7 に訓練セットのサイズを示す。実験結果を表 8 に示す。訓練コーパスを増やすことにより翻訳品質が向上していることが分かる。

表 9 は距離閾値が 1/4 の場合の結果を示す。表 8 と比較すると、当然ながら翻訳出力率は下がるが、処理時間の短縮は明瞭である。閾値を小さくして用例を増やすと、処理時間の増大を抑え、かつ翻訳出力率を落とさないことが可能である。このとき翻訳結果の得られた入力文の用例に対する距離は同じか小さくなり

表 7 訓練コーパスサイズ
Table 7 Training corpus size.

コーパスサイズ	19K	38K	75K	150K	300K	
のべ文数	19,022	38,043	76,085	152,170	304,340	
異なり文数	15,923	29,785	54,657	97,116	199,664	
一致テスト文 (のべ)	文数	104	142	181	227	240
	率 (%)	20.4	27.8	35.5	44.5	47.1

表 8 コーパスサイズ別日英翻訳性能 (距離閾値=1/3)
Table 8 Training corpus size and JE translation performance (distance threshold=1/3).

サイズ	mWER	BLEU	主観評価ランク (%)			出力率 (%)	処理時間 (秒)	
			A	AB	ABC		平均	最大
19K	0.4617	0.5153	53.7	65.9	71.0	82.5	0.062	0.550
38K	0.4136	0.5824	58.2	70.6	75.1	86.3	0.085	0.930
75K	0.3858	0.6157	64.7	74.1	79.0	88.8	0.121	1.690
150K	0.3425	0.6399	71.2	80.4	83.3	91.8	0.218	3.310
300K	0.3213	0.6331	71.2	81.8	84.5	93.3	0.320	6.650

表 9 コーパスサイズ別日英翻訳性能 (距離閾値=1/4)
Table 9 Training corpus size and JE translation performance (distance threshold=1/4).

サイズ	mWER	BLEU	主観評価ランク (%)			出力率 (%)	処理時間 (秒)	
			A	AB	ABC		平均	最大
19K	0.5080	0.4567	51.0	61.6	64.1	72.2	0.031	0.390
38K	0.4610	0.5268	55.1	65.9	68.8	77.6	0.041	0.680
75K	0.4174	0.6072	63.1	71.8	75.1	82.2	0.053	0.530
150K	0.3626	0.6608	70.2	78.4	80.2	86.5	0.096	1.600
300K	0.3364	0.6496	70.4	80.2	82.2	89.4	0.133	2.040

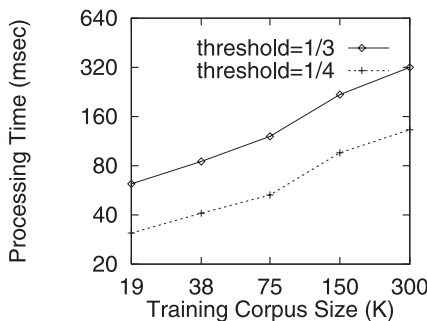


図 5 コーパスサイズと翻訳処理時間
Fig. 5 Training corpus size and processing time.

訳質向上につながる。つまり用例量と距離閾値の組合せによって訳質, 出力率, 処理速度が調整できる。たとえば, 表 9 のコーパスサイズ 300 K の結果は, 表 8 でのコーパスサイズ 150 K の結果に比べ, 訳質と出力率は若干劣化しているが 1.6 倍の処理速度を得ている。

図 5 はコーパスサイズと平均処理時間の関係を示す。縦軸, 横軸は対数スケールとなっている。コーパスサイズを大きくするに従って処理時間も大きくなっている。しかし増加の程度は線形比例ではなく, 処理

時間はコーパスサイズのほぼ 1/2 乗のオーダに抑えられている。

5.3.2 対訳辞書

対訳辞書の翻訳品質への影響を計測した。対訳辞書を, 1) ドメイン辞書, 2) 一般辞書, 3) 自動辞書, 4) 一般辞書+自動辞書, 5) 辞書なし, のように差し換えて翻訳を行った。実験用に準備したものの日英対訳辞書は, 旅行会話ドメイン用に開発された辞書と市販辞書から構成される。旅行会話ドメイン辞書が 51,607 エントリ, 市販辞書部分が 59,257 エントリを持つ。両者に共通するエントリではドメイン辞書の対訳を選択し全体としてのエントリ数は 93,848 となっている。1) では旅行会話ドメイン辞書のみ使用する。2) では市販辞書のみ使用する。3) では統計的手法による単語ライメント¹²⁾を利用して訓練セットから辞書を作成した。このエントリ数は 3,010 である。

実験結果を表 10 に示す。A ランクの割合の大きい順に上から並べてある。ドメイン辞書が一番良い結果が得られ, 辞書をまったく使わないと品質は大きく劣化する。A ランクが 15%程度低下する。なおテストセット 510 文のうち 468 文で翻訳結果が得られてい

表 10 対訳辞書別日英翻訳品質

Table 10 Transfer dictionary and JE translation accuracy.

対訳辞書	mWER	BLEU	主観評価ランク (%)		
			A	AB	ABC
1) ドメイン	0.3425	0.6412	71.2	80.4	83.3
4) 一般+自動	0.3530	0.6359	65.5	74.3	78.2
3) 自動	0.3532	0.6424	64.1	72.2	76.1
2) 一般	0.3606	0.6258	63.1	72.2	77.1
5) 無	0.3677	0.6583	55.7	65.7	70.8

表 11 ドメイン辞書サイズ別日英翻訳品質

Table 11 Domain dictionary and JE translation accuracy.

サイズ	mWER	BLEU	主観評価ランク (%)		
			A	AB	ABC
1,000	0.3480	0.6411	66.9	75.3	78.8
2,000	0.3473	0.6439	67.8	76.5	79.6
3,000	0.3472	0.6427	68.4	77.1	80.4
4,000	0.3466	0.6250	69.2	77.3	80.8
5,000	0.3465	0.6439	69.4	77.6	81.2
10,000	0.3460	0.6336	69.8	78.4	82.0

るが、ドメイン辞書使用時、翻訳出力文中の単語総数は 2,450 であり、そのうち対訳辞書により置換された部分は 260 語である。対訳辞書による置換語数は少ないが、対訳辞書の翻訳品質に与える影響は大きいといえる。

一般辞書と自動辞書は、異なるドメインへの移植時にコストをかけずに利用可能な言語資源と見ることができる。一般辞書と自動辞書を合わせて使う条件では、ドメイン辞書に比べてときの A ランクの低下幅が 5%程度にとどまっている。

次に、コストをかけてドメイン辞書を追加する場合を想定し、追加する辞書のサイズと翻訳品質の関係を観察する。ここでは、訓練セット内の日本語の出現頻度を基準にし、ドメイン辞書の高頻度エントリから取り出して作る辞書を 1,000 エントリ単位で増やしていく、一般辞書と自動辞書と合わせて使用した。表 11 の結果は、ドメイン辞書を増やすことによる着実な訳質向上を示している。

5.3.3 シソーラス

シソーラスの翻訳品質への影響を計測した。原言語のシソーラスは用例検索と翻訳パターン抽出フェーズで利用される。目的言語のシソーラスは翻訳パターン抽出フェーズで利用される。用例検索フェーズでは、シソーラスは用例選択基準に意味的要素を加味し、距離に細かな差をつける。翻訳パターン抽出フェーズでは、類語関係に基づいた単語間の対応関係の検出を可能とする。

ここではシソーラスを使う場合と使わない場合の翻

訳結果を比較する。原言語のシソーラスを使わなければ単語列編集距離が大きくなり翻訳出力率が低下する。これは意味距離を考慮しないように距離計測基準を変えたことに相当するので、同じ距離閾値を使って翻訳品質を比較することにはあまり意味がなくなってしまう。そこで距離閾値を調整し、シソーラスを使ったときと同程度の翻訳出力率が得られる場合の翻訳結果も比較対象とした。なお、シソーラスサイズは、原言語(日) 155,949 語、目的言語(英) 80,250 語である。

実験結果を表 12 に示す。シソーラスを使わなければ翻訳出力率が下がり翻訳品質が劣化する。しかし距離閾値を 0.41 に上げ翻訳出力率の低下を抑えると、シソーラスを使った場合に近い翻訳品質が得られる。処理時間を見ると、シソーラスを使わない場合は使う場合に比べ最大値が 10 倍以上になる。処理時間が大きくなる翻訳実行例では、用例検索時間は小さいが、最小距離の用例が非常に多数存在するため、翻訳パターン抽出フェーズ以降で時間がかかっている。これらの結果からシソーラスを使う効果は、翻訳品質に関してよりも、用例の絞り込みによる安定した処理時間の抑制効果にあるといえる。

5.4 多言語適用性

日英に加え、日韓、日中、英日、韓日、中日の各方向の翻訳を実施した。対訳辞書のエントリ数は、日韓 11,762、日中 38,126、英日 80,095、韓日 5,188、中日 28,691 である。韓日、中日の主観評価では、評価者に渡すデータの韓国語、中国語のテスト文を対応する英文に置き換え、英日翻訳として評価ランクが選択された。日韓、日中についての主観評価は行っていない。韓国語と中国語のシソーラスを使っていないため韓日翻訳、中日翻訳では距離閾値を上げることにより他言語対と翻訳出力率を同程度に揃えている。

表 13 に実験結果を示す。言語対によって本質的な翻訳の難しさ、言語資源の整備の状況はまちまちであるが、十分整備されている日英と比べうる翻訳品質が得られている。特に韓国語、中国語に関しては、与えられた対訳コーパスと対訳辞書を使い、その言語の知識を持つ人間の関与なしで機械的に翻訳システムを構築し評価している。言語資源を用意するだけで様々な言語対の翻訳システムが実現可能であることが確認された。

6. 考 察

5 章で示した大規模コーパスを使った実験結果をふまえて、翻訳システムの開発効率、翻訳カバレッジに関する課題について考察する。

表 12 シソーラス別日英翻訳性能
Table 12 Thesauri and JE translation performance.

シソーラス	mWER	BLEU	主観評価ランク (%)			出力率 (%)	処理時間 (秒)	
			A	AB	ABC		平均	最大
有	0.3425	0.6399	71.2	80.4	83.3	91.8	0.218	3.310
無	0.3772	0.6699	67.3	75.7	77.8	82.2	0.223	40.520
無 (閾値 0.41)	0.3377	0.6576	69.6	80.0	83.3	91.6	0.316	40.890

表 13 多言語翻訳品質
Table 13 Multilingual translation accuracy.

翻訳方向	mWER	BLEU	主観評価ランク (%)			出力率 (%)
			A	AB	ABC	
日英	0.3425	0.6399	71.2	80.4	83.3	91.8
日韓	0.3216	0.6426	-	-	-	91.8
日中	0.2898	0.6881	-	-	-	91.8
英日	0.3427	0.6767	58.0	70.6	78.0	92.0
韓日	0.3062	0.7101	55.1	61.6	69.0	92.2
中日	0.3208	0.7211	57.6	65.1	71.8	91.8

6.1 開発効率

D³ で翻訳システムを実現するには、対訳コーパス、対訳辞書、シソーラスの言語資源を必要とする。この資源を用意するためのコストを削減することが開発効率向上につながる。

この観点からまず対訳辞書について考察する。今日では言語対によっては電子的に利用可能な一般辞書も広く入手可能である。また対訳コーパスから辞書を自動作成することも可能である。これらの一般辞書や自動辞書で十分な翻訳品質が得られるならば、翻訳システム開発時に新たに人手をかける必要はない。実験結果は、対象ドメイン用に作成した辞書を使ったときの翻訳品質とは差があるものの良い翻訳品質を示している。さらに高い品質が望まれる場合は、システムに求められる品質に応じて人手をかけて対象ドメイン用の対訳辞書を作成すればよいことになる。また辞書の自動構築の手法に関しては多くの提案がなされており研究の地道な進展により自動辞書の有用性の向上が期待される。

シソーラスを準備するためのコストの削減には、シソーラスコードを自動付与したり²⁴⁾、類語関係をコーパスから学習したり¹⁾する手法が有望である。一方でシソーラスを使わない場合の翻訳品質の劣化が小さいことが実験により示されている。そのためシソーラスを使用しないという選択肢もある。ただしシソーラスを使わなければ、類似用例の絞り込みが進まず翻訳処理に時間がかかってしまう場合がある。そのため、候補の絞り込みのための新たな指標や効率的なアルゴリズムの導入が必要になる。

コーパスサイズを大きくすれば翻訳能力が向上し、

かつ大規模コーパスを使った場合でもリアルタイムの翻訳処理が可能であることが実験により確認された。処理時間はコーパスサイズのほぼ 1/2 乗のオーダに抑えられている。他方で対訳コーパス作成のためのコストも問題となるが、自動的に文アライメント¹¹⁾された対訳コーパスを利用することがコスト削減につながる。自動的に作成された対訳コーパスを用例として利用するにはコーパスの誤りに対する頑健性が特に重要になる。このためには翻訳処理に不適な用例を排除する手法¹⁰⁾の適用などが考えられる。また、少量の対訳コーパスしか準備できない場合を想定して翻訳品質を上げる手法も研究されている⁷⁾。この試みも対訳コーパスを準備するために必要なコストの削減につながる。

6.2 翻訳カバレッジ

多くの用例翻訳では、1つの文を翻訳するのに複数の用例から抽出した翻訳パターンを組み合わせている。D³についても複数の用例を組み合わせる研究が行われている⁸⁾。複数の用例を組み合わせることにより、より多くの文の翻訳が可能になる反面、接合部分の不整合により訳文に誤りや不自然さが加わる余地が大きくなる。本稿では複数の用例の組合せは行わず、文単位の用例1つで1つの訳文を構成する単純な条件下での翻訳品質について評価した。

結果として高い翻訳品質が得られる一方、翻訳処理に有効な近距離の用例が存在しないために翻訳不能となる場合もあった。今回の実験用のコーパスでは短い文が多いが、入力文が長くなれば翻訳できない可能性が大きくなる。翻訳可能な文のカバレッジを上げる1つの方法は、上述のように、文単位に限らず句単位の

用例も扱い入力文を分割して複数の用例を組み合わせる手法を導入することである。他の考え方は、 D^3 は低カバレッジ・高品質の翻訳方式だと割り切り、他の高カバレッジ・中品質の翻訳方式と相互に補完して使うことである。

7. おわりに

単語列編集距離を使った用例翻訳である D^3 の実装方式と大規模コーパスを使った評価実験について報告した。 D^3 では用例を抽象化せず単語列の形のまま保持し、翻訳実行時にも目的言語文の変更を最低限にとどめるため、入力文に十分近い用例がある場合は自然な翻訳文が生成される。また構文解析処理やツリーバンクを要しないため多言語適用性に優れる。

D^3 の実装にあたり、特に大量の用例を利用する場合を想定し、効率的な用例検索手法を提案した。評価実験では、言語資源と翻訳品質および処理時間の関係を分析し、また多言語適用性を実証した。特にコーパスサイズを大きくすれば翻訳品質が向上し、かつ大規模コーパスを使った場合でもリアルタイムの翻訳処理が可能であることが確認された。また対訳コーパス、対訳辞書、シソーラスの言語資源を用意することにより、様々な言語対の翻訳システムを実現することができた。これらの実験結果をふまえ、翻訳システムの開発効率を左右する要因および翻訳カバレッジの問題に関して今後の課題を考察した。

謝辞 類語新辞典の使用許可をいただいた角川書店様に感謝いたします。なお、本研究は情報通信研究機構の研究委託「大規模コーパスベース音声対話翻訳技術の研究開発」により実施したものです。

参考文献

- 1) Alshawi, H. and Douglas, S.: Speech Translation Performance of Statistical Dependency Transduction and Semantic Similarity Transduction, *Proc. 40th ACL workshop on Speech-to-Speech Translation*, pp.31–38 (2002).
- 2) Baldwin, T. and Tanaka, H.: Balancing up Efficiency and Accuracy in Translation Retrieval, *自然言語処理*, Vol.8, No.2, pp.19–37 (2001).
- 3) Brzozowski, J.A.: Canonical regular expressions and minimal state graphs for definite events, *Proc. Symposium of Mathematical Theory of Automata, MRI Symposia Series*, Vol.12, pp.529–561 (1962).
- 4) Carl, M.: Inducing Translation Templates for Example-Based Machine Translation, *Proc. 7th MT Summit*, pp.250–258 (1999).
- 5) Cormen, H.T., Leiserson, C.E. and Rivest, L.R.: *Introduction to Algorithms*, MIT Press, p.1028 (1989).
- 6) Cranias, L. and Papageorgiou, H.: Example retrieval from a translation memory, *Natural Language Engineering*, Vol.3, No.4, pp.255–277 (1997).
- 7) Doi, T., Sumita, E. and Yamamoto, H.: Adaptation Using Out-of-Domain Corpus within EBMT, *Proc. HLT-NAACL 2003*, Companion Volume, pp.16–18 (2003).
- 8) Doi, T. and Sumita, E.: Input Sentence Splitting and Translating, *Proc. workshop on Building and Using Parallel Texts, HLT-NAACL 2003*, pp.104–110 (2003).
- 9) Imamura, K.: Application of Translation Knowledge Acquired by Hierarchical Phrase Alignment for Pattern-based MT, *Proc. TMI-2002*, pp.74–84 (2002).
- 10) Imamura, K., Sumita, E. and Matsumoto, Y.: Automatic Construction of Machine Translation Knowledge Using Literalness, *Proc. EACL 2003*, pp.155–162 (2003).
- 11) Manning, D.C. and Schütze, H.: *Foundations of statistical natural language processing*, MIT Press (1999).
- 12) Melamed, I.D.: Models of Translational Equivalence among Words, *Computational Linguistics*, Vol.16, No.2, pp.221–249 (2000).
- 13) Nagao, M.: A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, *Artificial and Human Intelligence*, Elithorn, A. and Banerji, R. (Eds.), North-Holland, pp.173–180 (1984).
- 14) 大野 晋, 浜西正人: 類語新辞典, 角川書店 (1984).
- 15) Papineni, K., Roukos, S., Ward, T. and Zhu, W.: Bleu: a Method for Automatic Evaluation of Machine Translation, *RC22176, September 17, 2001, Computer Science* (2001).
- 16) Planas, E. and Furuse, O.: Formalizing Translation Memories, *Proc. 7th MT Summit*, pp.331–339 (1999).
- 17) Rapp, R.: A Part-of-Speech-Based Search Algorithm for Translation Memories, *Proc. LREC 2002*, pp.466–472 (2002).
- 18) Sato, S. and Nagao, M.: Toward Memory-based Translation, *Proc. COLING'90*, pp.247–252 (1990).
- 19) Sato, S.: CTM: An Example-Based Translation Aid System, *Proc. COLING '92*, pp.1259–1263 (1992).
- 20) 菅谷史昭, 竹澤寿幸, 横尾昭男, 山本誠一: 音声翻訳システムと人間との比較による音声翻訳能力

- 評価手法の提案と比較実験, 電子情報通信学会論文誌 D-II, Vol.J84-D-II, No.11, pp.2362-2370 (2001).
- 21) Sumita, E. and Iida, H.: Experiments and Prospects of Example-based Machine Translation, *Proc. 29th Annual Meeting of ACL*, pp.185-192 (1991).
- 22) 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, *Proc. 7th MT Summit*, pp.229-235 (1999).
- 23) Sumita, E.: Example-based machine translation using DP-matching between word sequences, *Proc. 39th ACL workshop on DDMT*, pp.1-8 (2001).
- 24) 鷹尾和亨, 柏岡秀紀, 白井 諭: 異なる辞書を利用した意味コードの自動付与, 情報処理学会第59回全国大会講演論文集, 1N-07 (1999).
- 25) Takezawa, T., Sumita, E., Sugaya, F., Yamamoto, H. and Yamamoto, S.: Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world, *Proc. LREC 2002*, pp.147-152 (2002).
- 26) Ueffing, N., Och, F.J. and Ney, H.: Generation of Word Graphs in Statistical Machine Translation, *Proc. Conf. on Empirical Methods for Natural Language Processing*, pp.156-163 (2002).
- 27) Veale, T. and Way, A.: Gaijin: A Bootstrapping, Template-Driven Approach to Example-Based MT, *Proc. NeMNL'97, New Methods in Natural Language Processing* (1997).
- 28) Watanabe, H. and Takeda, K.: A Pattern-based Machine Translation System Extended by Example-based Processing, *Proc. 36th ACL and 17th COLING*, pp.1369-1373 (1998).
- 29) Yamamoto, K., Shirai, S., Sakamoto, M. and Zhang, Y.: Sandglass: Twin Paraphrasing Spoken Language Translation, *Proc. ICCPOL*

2001, pp.154-159 (2001).

(平成 15 年 10 月 10 日受付)

(平成 16 年 4 月 5 日採録)



土居 蒼生 (正会員)

1984 年京都大学理学部卒業。同年 (株)CSK 入社。2002 年より ATR 音声言語コミュニケーション研究所に出向。自然言語処理 (特に機械翻訳), 自動推論システムおよびプログラミング言語の実装方式の研究に従事。技術士 (情報工學部門)。人工知能学会, ACL 各会員。



隅田英一郎 (正会員)

1982 年電気通信大学大学院計算機科学専攻修士課程修了。1999 年京都大学工学博士。現在, ATR 音声言語コミュニケーション研究所主任研究員。自然言語処理 (特に, 機械翻訳, 統計的文脈処理, 音声言語統合処理, 言語データベース, 並列自然言語処理), 情報検索, eラーニングの研究に従事。電子情報通信学会, 言語処理学会, 日本音響学会, ACL 各会員。



山本 博史

1979 年東京大学農学部農業生物学科卒業。1981 年同大学大学院農学系修士課程修了。同年 (株)CSK 入社。1996 年より ATR 音声翻訳通信研究所, ATR 音声言語通信研究所, ATR 音声言語コミュニケーション研究所に出向。国際情報通信学博士。音声認識の研究に従事。電子情報通信学会, 言語処理学会, 日本音響学会各会員。