修 士 論 文

構造的言語処理を指向する
用例ベース機械翻訳システム

指導教員　　江崎 浩 教授

東京大学大学院
情報理工学系研究科 電子情報学専攻

学籍番号・氏名　56430　中澤 敏明

提 出 日　　平成18年2月2日

目 次

概要

近年の計算機パワーの増大や、他言語に触れる機会の増加などにより、機械翻訳技術への期待が高まっている。しかしながら、実用レベルのシステムはいまだに構築されておらず、研究の余地が多く残されている分野である。

本論文では、代表的な機械翻訳手法の一つである用例ベース翻訳手法を用いた機械翻訳システムの構築を目指す。機械翻訳は大きく分けて、翻訳するための知識獲得をするアラインメント部と、獲得した知識を利用して入力文を翻訳する翻訳部に分けられる。

ここで重要なのがアラインメント部であり、いかに過不足なく、誤りの少ない翻訳知識を獲得できるかが、翻訳の精度に大きく影響する。そこで我々は、アラインメントの精度を向上するための新たな手法の提案も行なう。この手法を適用することにより、アラインメントの精度をベースライン手法に比べて3ポイント以上向上することに成功した。

第1章　はじめに

機械翻訳の歴史は古く、1940 年代後半から始まったと言われている。それにもかかわらず、現時点ではまだ高精度の機械翻訳システムは実現されていない。この要因としては、機械翻訳には大きな計算機パワーが必要であるここと、大規模な対訳データや高度な言語処理技術などの言語資源・言語知識が必要であることが挙げられる。しかしこれらの問題は、近年の計算機のめざましい発達や、インターネットなどによる対訳データの利用、さらには言語処理技術の発展などにより解決されつつある。それとともに、他言語に触れる機会も急速に増加しており、機械翻訳への期待が高まっている。

1.1　これまでの機械翻訳研究

機械翻訳は 1940 年代から研究が始まったとされており、そこには現在の統計翻訳に通じるアイデアがある。その基本的なアイデアは、1947 年にロックフェラー財団の Warren Weaverが送った次のような手紙 [19] にその端を発する。

> ロシア語の文章を見たとき、私は言った「これは、本当は英語で書かれたものだが、変な記号に暗号化されている。今からデコードを行おう。」

ここで表現されているように、翻訳は暗号解読と同じ手法で行うというアイデアに基づいている。ここでいう暗号解読とは観察された記号列 (原言語の文) を一番もっともらしいもとの記号列 (目的言語の文) にデコードするという操作である。

最初の機械翻訳では翻訳の規則を人手により書き下して翻訳する手法が用いられ、**ルールベース翻訳 (Rule-based Machine Translation:RBMT)** と呼ばれる。ルールベース翻訳は現在でも商用翻訳アプリケーションの主流として利用されている。しかし当時の計算機では、その性能の制約から単語レベルの翻訳しかできず、文の生成にはいたっていない。このような状況がしばらく続き、さらには「機械翻訳には多義性解消などの高度な知識処理が不可欠である」という意見が出されるなど、機械翻訳は不可能であると考えられるようになり、機械翻訳の研究は一度は停滞期を迎える。

1970 年代に入ると言語処理研究も進み、これにより機械翻訳の研究が再開される。言語理解が進むにつれて、翻訳の精度やレベルも徐々に向上していった。1981

年には長尾 [13] によって、**アナロジーに基づく機械翻訳**が世界ではじめて提唱され、後の用例ベース翻訳の基礎を築いた。

1982 年には「科学技術庁機械翻訳プロジェクト (Mu プロジェクト)」が発足し、現在市販されている日本の商用機械翻訳システムに大きな影響を与えた。このプロジェクトでは、諸外国との科学技術文献交流促進の必要性から、それらの文献 (抄録) の翻訳を効率的に行なう目的で、日英／英日翻訳システムの開発を行なった。

その後計算機の高性能化や、対訳コーパスの充実などにより、大量のデータから翻訳を統計的に学習する**統計翻訳**の研究が行なわれるようになる。代表的なものには Brown ら [2] によるものがある。現在ではこの統計翻訳と用例ベース翻訳が主流となって、活発に機械翻訳の研究が行なわれている。しかし現在でも高精度な機械翻訳システムの開発は実現されておらず、今後も研究する余地が多く残されている。

1.2 機械翻訳の代表的手法

機械翻訳の代表的な手法には、大きく分けてルールベース、統計、用例ベースの 3 つがある。前章でも触れたように、ルールベース翻訳は現在でも商用のシステムで用いられているが、翻訳の規則を一つ一つ人手で書き下す必要があり、メンテナンスの上でも、また多言語対応するための手間の上でも、不利であるといえる。また、そもそも全ての言語現象を逐一書き下すことは不可能であり、ルールベース翻訳ではロバストで高精度な翻訳システムの構築を望むのは難しい。

そこで盛んに研究されているのが、統計翻訳と用例ベース翻訳である。これらの手法は、与えられた対訳コーパスから翻訳に用いる知識を学習して入力文の翻訳を行なうため、メンテナンスも容易であり、ロバストな手法であるといえる。それぞれの手法の特徴を以下に述べる。

1.2.1 統計翻訳

統計翻訳では言語資源が対訳コーパスしかなく、対訳辞書や言語知識を用いずに翻訳するという問題設定で翻訳を行なう。それゆえ、単語などの小さな単位で翻訳を行なうことが基本的な方法である。しかし最近では、単語列や句などのより大きな単位を扱ったり、汎化した単位を扱ったりして統計翻訳を行なうことも多い。また構文情報を利用した統計翻訳も登場しており、純粋に対訳コーパスのみからの翻訳を行なう研究は少くなってきている。

統計翻訳のモデル

統計翻訳では、たとえば日本語文 J から英語文 E への翻訳は、以下の条件付き確率の最大化で表現される。

$$\begin{aligned} E &= \arg\max_E P(E \mid J) \\ &= \arg\max_E P(E)P(J \mid E). \end{aligned}$$

上式で $P(E)$ は**言語モデル**と呼ばれ、目的言語文 (ここでは英語文) のもっともらしさを表わすモデルである。$P(J \mid E)$ は**翻訳モデル**と呼ばれ、ある目的言語の文 (ここでは英語文) が与えられたときに、その翻訳として、ある原言語の文 (ここでは日本語文) が生成される確率を表わすモデルである。翻訳を実現するには、この二つのモデルが必要となる。

翻訳モデルの構成と学習

翻訳モデルは、統計翻訳の研究の中でも活発に研究が行なわれているものの一つである。数多くある翻訳モデルの中でも、もっともよく利用されるものに、IBM Model 4 と呼ばれるモデルがある。ここでは、この IBM Model 4 について述べる。

IBM Model 4 では、翻訳モデルを以下の 4 つのモデルに分割して考える。

Fertility モデル ：英語の各単語が生成する日本語単語の数。

NULL generation モデル ：生成する文の長さをあわせるために、NULL を生成するモデル。

Lexicon モデル ：ある英単語がある日本語単語に翻訳される確率。1 対 1 の単語単位で考える。

Distortion モデル ：翻訳における語順の変化を表現するモデル。

もし図 1.1 のように、ある対訳文について単語同士の対応付け (アラインメント) が正確に与えられているならば、それらを計数することによって、これらのモデルを構築することは容易である。しかし一般的にこのようなアラインメント情報を大量に人手によって付加することは難しく、教師あり学習をすることは不可能と言ってよい。そこで EM アルゴリズムなどにより、教師なし学習でパラメーターの推定を行ないつつ、モデル学習を行なうという手法が取られることが多い。

1.2.2 用例ベース翻訳

用例ベース翻訳では、対訳文をそのまま、もしくはある程度抽象化したものを翻訳知識 (用例) とし、入力文にできるだけ近い用例を使って翻訳する。

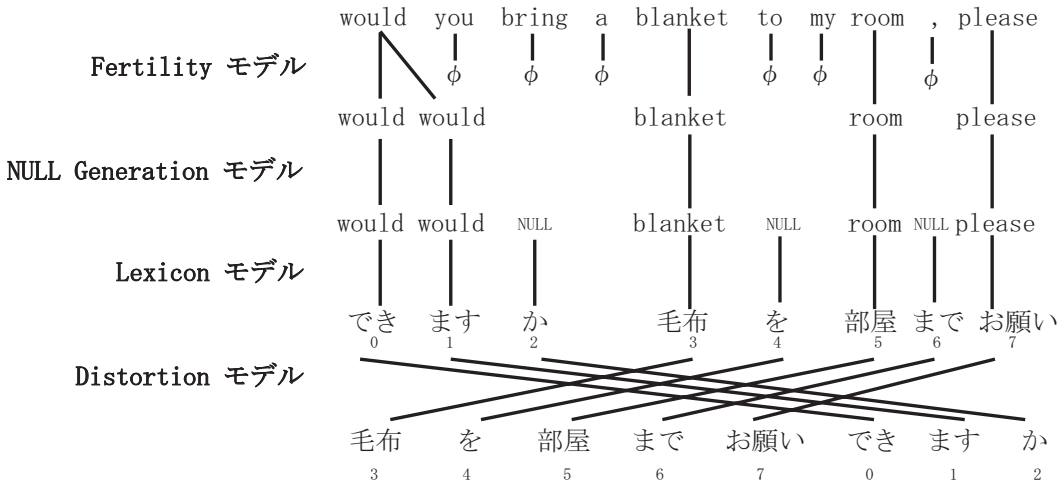


図 1.1: 翻訳モデル

統計翻訳で安定した翻訳結果を得るためには、大規模な対訳データからの学習が必要である。一方用例ベース翻訳では、小規模な対訳データしか利用できない場合であっても、ドメインの近い文章の翻訳ならば、似たような用例を用いることによって安定した翻訳を得ることや、翻訳支援をすることが可能である。

用例ベース翻訳の基本的なアイデアは、入力文をいくつかの部分に分解し、その部分ごとに類似した用例を用いて翻訳を行い、それらを組み合わせるというものである [13]。

例えば、図 1.2 の例では、入力文 “毛布を部屋までお願いできますか” に対して、“部屋までお願いできますか”、“毛布は” の 2 つの用例を利用し、これらの組み合わせによって翻訳文を作り出している。

対訳データを用いるという点、また、それらが内部でアラインメントされている必要があるという点は統計翻訳と共通である。しかし、統計翻訳が、文を単語に分解し、できるだけ頻度の高いものの組み合わせを優先しようとしているのに対し、用例ベース翻訳は、できるだけ大きな用例の組み合わせを優先しようとする点が異なる。また、同じ大きさの用例の場合には、さらにその外側の表現が類似している方がよいという尺度を考える。このようにして、できるだけ大きな文脈で用例を用いることによって正確な翻訳を得ようとするのである。

用例ベース翻訳の初期の研究としては名詞句「A の B」の訳し分けを扱った隅田らの研究 [17]、佐藤による動詞と必須格の訳語選択を行う研究 [21] や、複数用例の組み合わせを議論した研究 [22] などがあるが、扱う表現のバリエーション、データ量などは小規模なものであった。しかし、これも 90 年代後半からの対訳データの規模の拡大にともない徐々に大規模なシステムの構築が行われるようになっていった。用例ベース翻訳の最近の成果については Andy らの論文 [3] によくまとめられている。

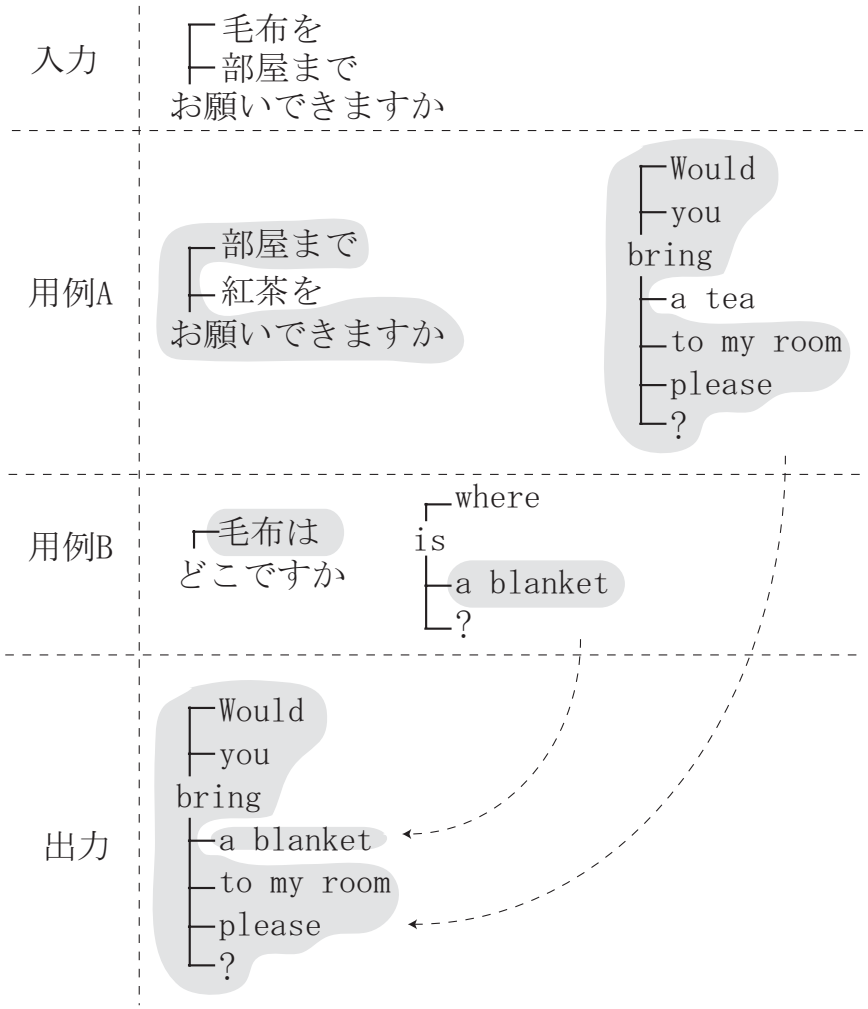


図 1.2: 用例ベース翻訳の概要

1.3 用例ベース翻訳の有用性

現在の機械翻訳の主流は、Brown ら [2] や Och ら [14] に代表されるように統計翻訳であるが、我々は用例ベース翻訳システムを研究している。これには以下の 2 つの理由がある。

1 つは、構造的な言語処理の発展を指向しているからである。機械翻訳は、言語処理研究の成果である形態素解析や構文解析などの基礎技術のアプリケーションとしてとらえることができる。つまり、基礎技術の発展がアプリケーションの精度向上につながるであろうし、逆にアプリケーション側から、基礎技術の弱点が見えてくることもある。

2 つ目は、用例ベース翻訳の問題設定が妥当である場合が少くないからである。たとえば、マニュアルのバージョンアップの翻訳や、関連特許の翻訳などは、コーパスの規模としてはそれほど多くなく、統計翻訳がうまく働くかというと疑問が残る。しかしこのような場合、ドメインが同じならば、同じような用例が多く得られるはずであり、用例ベース翻訳の考え方が有利に働くものと考えられる。

このような観点から、我々は用例ベースの手法を選択した。機械翻訳システムは大きく分けて**アラインメント部**と**翻訳部**に分けられる。我々が構築した機械翻訳システムのアラインメント部は 3 章で、翻訳部は 4 章で詳述する。

第2章　高度な日本語処理技術

高精度な機械翻訳システムを構築するためには、様々な言語処理技術を用いる必要がある。形態素解析や構文解析などの基礎技術はもちろんのこと、それらを応用した様々な技術を利用することにより、翻訳の精度を向上させることができるはずである。

ここでは、我々の機械翻訳システムに統合した二つの高度な言語処理技術について述べる。一つは日本語の柔軟マッチングであり、もう一つは省略された代名詞の推定手法である。

2.1　日本語の柔軟マッチング

2.1.1　日本語の表現の自由度

自然言語は自由度が高いため、同じ内容を表現するにしても様々な表現を使用することができる。そのような様々な表現のずれをいかにして吸収するかが自然言語処理における重要な課題の一つである。

これは機械翻訳の場合も例外ではなく、例えば入力文として “ホテルに一番近い駅はどこですか”が与えられたとする。このとき、用例に “旅館の最寄りの駅はどこですか ↔ Where’s the nearest station from the hotel?” があったとしても、単純な完全マッチングを行うだけではこの用例を翻訳に使うことはできない。

我々はこのような問題を克服するために、大西らの手法 [23] を取り入れた。

大西らは、表現のずれを吸収する柔軟なマッチングを実現するために、同義・類義表現の知識の獲得と、それらを柔軟に統合して利用する枠組みの 2 つを提案した。

1. 国語辞典から自動的に同義関係や上位下位関係の知識を獲得
2. 表現のずれを効率的に扱える SYNGRAPH データ構造を導入

1 については、国語辞典を用いることで、常識的、基本的な同義・類義表現を網羅的に、完全に自動的に獲得する。これにより、“夕食” と “食事” のような上位関係、“旅館” と “ホテル” のような単純な同義語関係に加えて、“一番” と “もっとも” などの副詞の同義語や、“最寄り” と “一番近い” のような語と句の同義関係など、広い範囲の類義関係を扱うことができる。

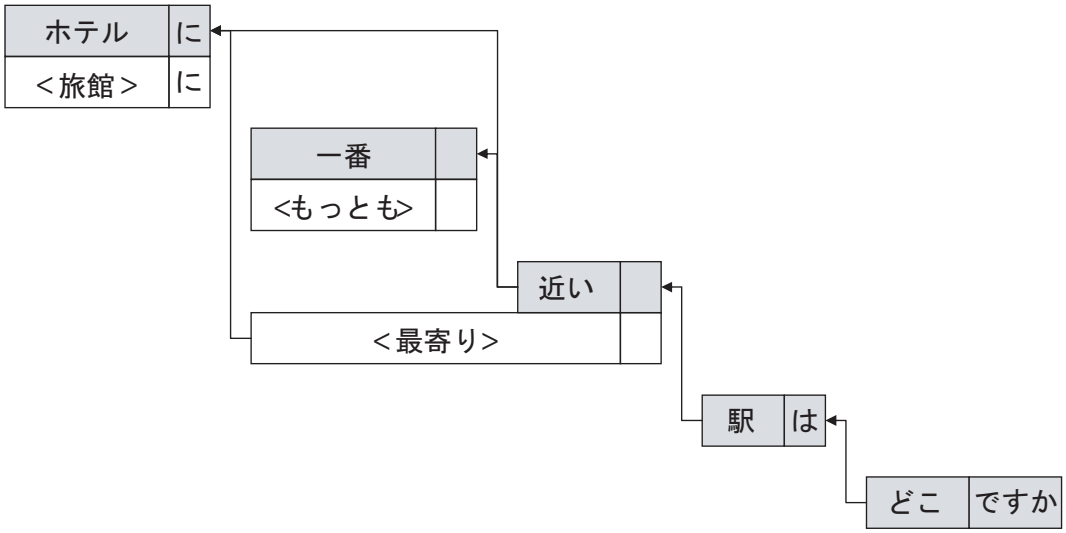


図 2.1: SYNGRAPH の例

2 については、類義関係を事前にすべて展開することは組合せ爆発となってしまい、ダイナミックに検索・マッチングを行う手法では計算量が大きすぎる。そこで、同義関係に ID を与え、表現のずれをまとめて扱える SYNGRAPH というデータ構造で文を表現する。

2.1.2 国語辞典の利用

同義関係や上位下位関係の知識源としてシソーラスがあるが、既存のシソーラスは目標とする柔軟マッチングには不適切である。その理由の一つは、シソーラス上で一つの意味素に属する語の数が多すぎるため、細かい意味の違いで同義表現を区別することが難しいからである。入力文と最も近い用例を翻訳に利用する必要がある用例ベース機械翻訳においては、この細かい意味の違いを正確に扱うことが非常に重要である。またシソーラスには単語と句の間の関係が記述されていることが少ないことも問題である。

これらの問題を解消するために、大西らはシソーラスの代替として国語辞典を使うことを提案した。これにより前章で例に出した上位下位関係や、様々な同義表現、単語と句の間の同義表現を抽出することができる。

2.1.3 SYNGRAPH のデータ構造

ある文について、前節で抽出した類義表現を組合わせ的に使えば、様々な別の、類義の文を作り出すことができる。

ホテルに一番近い駅はどこですか
＝ ホテルにもっとも近い駅はどこですか
＝ ホテルの最寄りの駅はどこですか
＝ 旅館に一番近い駅はどこですか
ホテルに近い駅はどこですか
...

柔軟なマッチングで必要なことはこれらの類義関係を認識することであるが、これを事前に展開しておく方法は組合わせ爆発を起こしてしまい、ダイナミックに検索する方法では計算量が爆発する。

そこで、文のあらゆる類義表現をまとめた(あらゆる類義表現を生成できる)SYNGRAPHというデータ構造を考え、これを用いてマッチングを行うことによって柔軟なマッチングを実現する。上記の例を SYNGRAPH で表現したものを図 2.1 に示す。

SYNGRAPH のベースとなるのは、もとの文の依存構造木であり、そのノードは 1 つの自立語と 0 個以上の付属語からなる。各ノードや、複数ノードの組み合わせに対し、類義表現を再帰的に付加していくことにより、あらゆる類義表現を生成できる SYNGRAPH 構造を作ることができる。

2.1.4 SYNGRAPH マッチング

2 つの SYNGRAPH は、もとの文を過不足なくカバーする同一のノード群が、同一の係り受け関係をもつ場合にマッチすると考える(図 2.2)。マッチしているそれぞれのノード間にノードマッチスコア(NMS)を定義し、次に SYNGRAPH マッチスコア(SMS)を NMS の和として定義する。

2 つの SYNGRAPH がマッチするかどうかは、それぞれの SYNGRAPH のヘッドから、マッチするノードの係り受け関係をそれぞれたどっていくことで調べられる。また、2 つの SYNGRAPH のマッチには、複数のノードの対応付けが考えられる場合があるが(図 2.2 で上位の＜旅館＞ノードを対応付ける場合など)、その中から最も SMS が大きい対応付けを選択する。これは単純には組合わせ爆発を起こすが、実際にはヘッドからノードのマッチを調べていく際に DP によって効率的に探索することができる。

2.1.5 機械翻訳での利用

用例ベースの機械翻訳では、与えられた入力文の各部分について、全く同じ、またはほぼ一致する用例を組み合わせることで翻訳を行う。このとき、最初の例でも示したように、同義・類義表現の用例を用いるために柔軟マッチングが必要となる。入力文と用例のそれぞれを SYNGRAPH 化し、SYNGRAPH マッチを行うこ

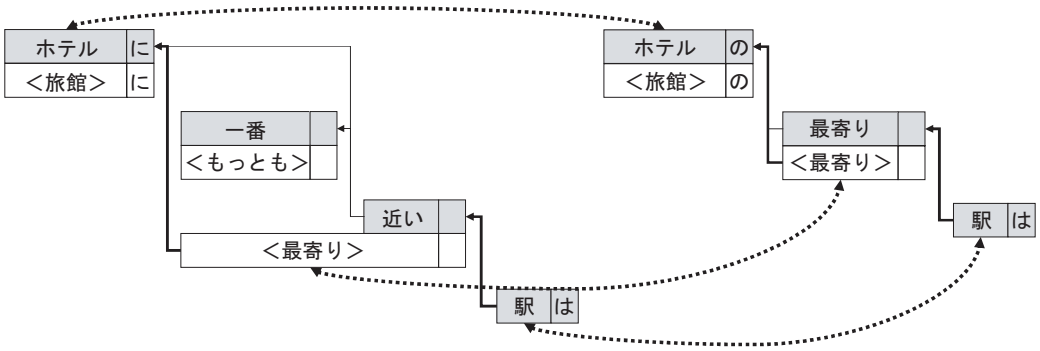


図 2.2: SYNGRAPH マッチングの例

とにより、柔軟に、かつ高速に用例を抽出することができ、ロバストな翻訳システムを構築することが可能である。

さらに対訳辞書を SYNGRAPH 化することにより、アラインメント部においても柔軟なマッチングを利用することができる。これについては 3 章で述べる。

2.2 人称推定

日本語は、英語などの西欧諸語とは対照的に、聞き手に推測できる格要素は表現しない言語である。特に対話文においてはそれが顕著であり、日本語の対話では “わたし” や “あなた” などの話し手聞き手を直接示す語はほとんど登場しない。それでも問題なく互いの発言を理解できるのは、聞き手が欠けている情報を推測し補っているからである。しかし、この人間にとっては互いに自明な省略も、機械にとっては必ずしも自明ではなく、このことが、日本語を様々な言語に機械翻訳する際に障害となってしばしば現われる。例えば

何を飲みますか。

という発話では “誰” が飲むのかという情報 (格要素) が省略されているが、日本語話者ならば誰でもそれが、

あなたは何を飲みますか。

という意味であり、“あなた” が省略されていると理解できる。また理解できるからこそ通常の対話では省略される要素となる。しかし英語では、

What do you drink ?

のように表現し、この文の “you” は絶対に省略できない。このため、最初の例文を英語に機械翻訳する際には、“you” を補わなければならず、そのためにまず “あなた” が省略されていることを機械が判断しなければならないことになる。しかし、

それは決して容易なことではない。省略されている格要素の判断には、平叙文や疑問文などの文の形から、動詞の持つ意味、さらに文が用いられたときの状況や話者の態度まで、様々な要素が絡んでくる。人間にとって自明な格要素を機械に推定させることは非常に困難な問題である。

そこで我々は、岡嶋らの手法 [20] を取り入れ、省略された格要素の人称推定を行なった。

岡嶋らは、モダリティと呼ばれる文中の要素を手がかりにして、この省略された格要素の人称を推定する手法を提案した。モダリティとは、話者の心的態度を表わすような一定の語や表現のことであり、疑問を表わすモダリティや依頼を表わすモダリティなど、多くのモダリティが研究されている。これらのモダリティを持つ文は意味的に制限され、格に登場する人称にも制限や偏りが生じる。このことを利用して省略された格の人称を推定する。

2.2.1 推定する格要素

岡嶋らは人称推定の対象として**格助詞格**、**モダリティ主体**、**主題格**の 3 つを用いた。これらの格要素に対して、一人称、二人称、三人称などの人称を推定する。

格助詞格

表層格とは、文中に表層的に現われた形で判断される構文的な格の分類であり、日本語においては名詞に付随する助詞の種類で分類される。

例えば、

> わたしが彼に話します。

という文の場合、動詞 “話す” の表層格は、ガ格が “わたし” で、ニ格が “彼” であるということになる。これらのうち、格助詞 (ガ、ヲ、...) で示されているものを特に格助詞格と呼んで、人称推定の主な対象とする。

主題格

対話文では、平叙文か疑問文かによってハ格に登場する人称に偏りが存在する。

a. 学校に行きます。(= 私は学校に行きます。)
b. 山が見えます。(= 私には山が見えます。)
c. 学校に行きますか。(= あなたは学校に行きますか。)
d. 山が見えますか。(= あなたには山が見えますか。)

これらの文でカッコで示されているのが、発話を自然に解釈したときに聞き手が補って推測するだろう内容である。このように、ガ格やニ格など格助詞格としては一定しないが、平叙文か疑問文かでハ格の人称に一定の傾向がある。これは、

- 平叙文は、自分が知っている情報を伝える文であるから (=話し手が判断主体であるから)、話し手自身のことを話しているのだろう
- 疑問は、自分が知らないことを質問する文であるから (=聞き手が判断主体であるから)、聞き手のことを聞いているのだろう

という二つの推論が成り立つからだと考えられる。このようにして生じる人称の偏りを表現するために、ハ格を特に主題格と呼び、推定の対象とする。

モダリティ主体

日本語の文中には現われないが、モダリティで仮想的に表現されている主体をモダリティ主体と呼ぶ。モダリティ主体には**希求主体**と**判断主体**の二種類を考える。

希求主体は主に「疑問のモダリティ」および「依頼のモダリティ」が存在するときに現われる主体であり、かならず話し手 (一人称) を指す。

a. 彼はもう行きましたか? (I wonder he has gone.)
b. それをください。 (I want that one.)

といった疑問の文、依頼の文には、それぞれ「返答を聞きたいと願う話者」「聞き手の行動を願う話者」が暗に存在している。この話者を希求主体と呼び、文の情報として保持しておくことで、日英アライメントを円滑に行なう助けとすることができる。

判断主体は、「平叙のモダリティ」「疑問のモダリティ」を持つ文に対し想定される主体である。「平叙のモダリティ」があれば判断主体は話し手 (一人称)、「疑問のモダリティ」があれば判断主体は聞き手 (二人称) となる。

a. 彼はもう行ったようです。 (I think he has gone.)
b. 彼はもう行ったようですか。 (Do you think he has gone?)

上の文のような平叙文では、判断をしているのは話し手である。一方、下の文のような疑問文では判断をするのは聞き手である。この判断する主体を判断主体と呼び、希求主体と同様、文の属性の一部として保持する。

2.2.2 機械翻訳での利用

代名詞の省略が問題となるのは 2 つのパターンがある。ひとつは、用例の日本語側で代名詞が省略されていて、入力文では省略されていない場合である。この場

合、英語の代名詞が用例に含まれ、この用例が入力の翻訳に用いられると、入力文の代名詞の訳出と重複する。たとえば “胃が痛いのです ↔ I ’ve a stomachache” が用例であり、これを用いて “私は胃が痛いです” を訳する場合には単純には “I I ’ve a stomachache” となる。

逆に、用例には代名詞があり、入力文で省略されていると、代名詞のない翻訳が生成される。たとえば用例が “これを日本へ送って下さい ↔ will you mail this to Japan” で “日本へ送って下さい” を訳す場合はこの用例の一部で “will you mail to Japan” となる。

人称推定を行うことにより、これらの問題が解消される。省略された代名詞が推定されたら、それを元の日本語文に補い、アラインメント及び翻訳を行なう。これにより、日英対訳文間の情報の過不足が解消され、アラインメント、翻訳ともに精度の向上が見込まれる。

具体的な利用法については、3 章および 4 章で説明する。

第3章　対訳文アラインメント

ここからは、我々が構築した用例ベース機械翻訳システムについて述べる。システムは**アラインメント部**と**翻訳部**からなる。

アラインメント部では、翻訳部で利用するための翻訳知識を学習する。具体的には、対訳文中で各言語の対応する部分の推定を行ない、対応する部分を翻訳で用いる用例として、用例データベースに保存する。

日英対訳文アラインメントは、両言語のパーサと対訳辞書などを用いて、以下のステップからなる (図 3.1 参照)。

1. 日本語文と英語文の依存構造解析
2. 対応候補の探索
3. 対応候補の選択
4. 未対応部分の推定
5. 用例データベースの構築

以下、各処理の詳細を説明する。

3.1　日本語文と英語文の依存構造解析

日本語文の解析には、日本語の形態素解析システム JUMAN[10]、依存構造解析システム KNP[9] を用いる。これらは日本語文の構造を非常に高精度に解析することができ、新聞ドメインでは、形態素解析が 99%、構文解析が 90%の精度である。これらの解析結果から、日本語の依存構造が得られる。

日本語の依存構造の単位 (ノード) は、各自立語が 1 ノードとなるもので、助詞、接辞、助動詞などは自立語のノードにまとる。

英文については、まず Charniak の nlparser[5][4] を用いて句構造に変換する。この結果、各単語や句にタグが付与されるので、この情報を元に、句のヘッド (文の中心となる部分) を定義するルールを人手で作成して依存構造に変換する。

英語の依存構造の単位 (ノード) は、日本語と同じく、各自立語が 1 ノードとなるもので、前置詞や助動詞は自立語のノードにまとめた。

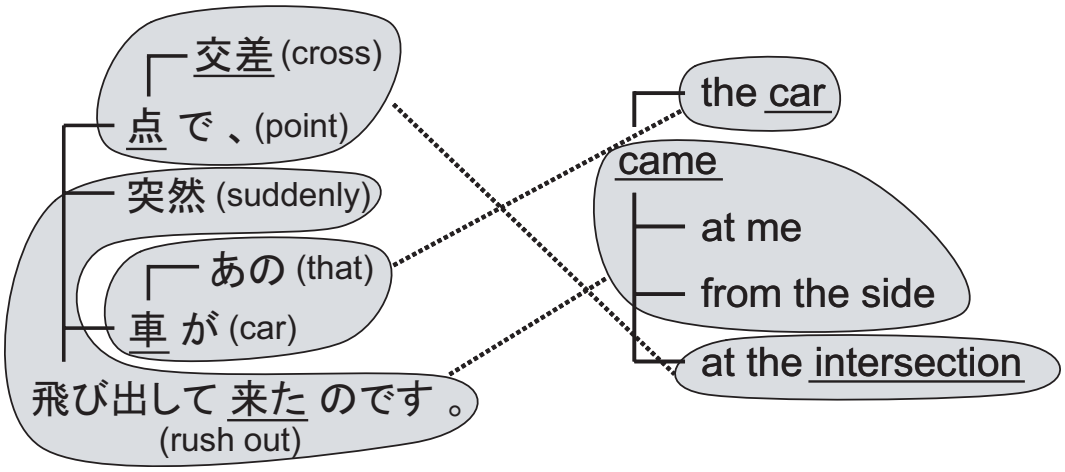


図 3.1: 対訳文アラインメントの例

なお日本語文に対しては構造解析と同時に、2.2 章で述べたように省略された代名詞の推定が行われており、代名詞の省略が検知された場合には、それを新たなノードとして追加している。この追加ノードは、他のノードと全く同等に扱う。

3.2 対応候補の探索

次に、日本語と英語、それぞれの単語・句対応の候補を探索する。対応の最小単位はノードとし、ノードの付属語は自立語の対応に含めて考える。ただし、複数自立語の対応が見つかった場合には、複数ノードの対応とする (図 3.1 の例では “交差 点で” と “at the intersection”)。

以下に対応候補の手がかりとして利用するものを挙げる。

3.2.1 対訳辞書

日本語と英語の 1 対 1 対応の辞書を用いる。1 単語だけではなく、複数単語にわたるエントリーも用いる。またこの辞書は確率的なものではない。

3.2.2 柔軟マッチング

自然言語は自由度が高いため、同じ内容を表現するにしても様々な表現を使用することができる。例えば、“最寄り” と “一番近い” という表現は同じ内容を表しており、英語では “nearest” となる。対訳辞書に “最寄り ↔ nearest” と “一番近い ↔ nearest” の両方が記述されていれば問題はないが、必ずしも記述されているとは限らない。

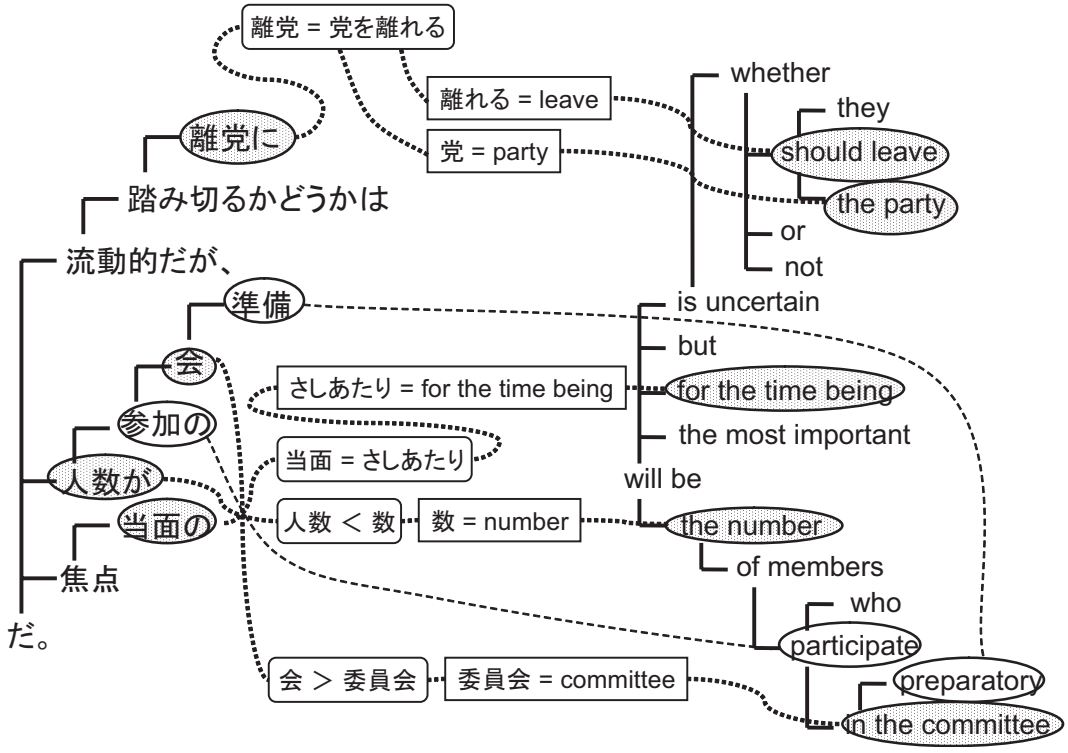


図 3.2: SYNGRAPH を利用した対応候補探索の例

そこで 2.1 章で述べたように、辞書を SYNGRAPH 化することにより柔軟なマッチングを行なう。図 3.2 に例を示す。

対訳辞書に含まれるエントリーをそのまま利用するだけだと、“参加 ↔ participate” や “準備 ↔ preparatory” といった少数の対応候補しか見つからないが、柔軟マッチングを利用することにより、さらに以下のような対応候補を発見することができる:

- “離党” の定義文が “党を離れること” であり、そこから “党 ↔ party”、“離れる ↔ leave” の対応が得られる。
- “当面” の類義表現である “さしあたり” と “for the time being” との対応が得られる。
- “人数” の上位語である “数” と “number” との対応が得られる。
- “会” の下位語である “委員会” と “committee” との対応が得られる。

3.2.3 数字の汎化

日本語の漢数字 (四十五や四五など) や英語の数字表現 (“forty five” や “forty fifth” など) をすべて算用数字 (45 など) に変換し、対応をとる。さらに数字の部分を<num>などに汎化することにより、数字が異なった場合でも翻訳時に用例として利用できるようにしておく。

日本語では数字の表記の仕方に種類はないが、英語には以下のように様々な種類があるため、汎化するときにどのタイプの数字表現なのかを記憶しておき、翻訳時にタイプに合うように復元する (4.5.1 章参照):

- 普通の表現: 124 ↔ one hundred (and) twenty four
- 2 桁ずつ (部屋番号、年号など): 124 ↔ one twenty four
- 1 桁ずつ (飛行機便名、電話番号など): 124 便 ↔ one two four
- 序数 (日にちなど): 2 日 ↔ second
- その他 (月など): 8 月 ↔ August

3.2.4 Transliteration

固有名詞、すなわち形態素解析結果から人名・地名であると判断された語と、カタカナ語 (外来語に使われることが多い) について、それらの可能性のある英語綴りの候補を自動で生成し、これらと、英語文中にある任意の英単語列との類似度を編集距離に基づいて計算する。類似度が閾値以上となる英単語列があれば対応候補とする。

例えば “新宿” という語に対して、まず “shinjuku” や “cinjyuku” など、考えうる英語表記を全て生成する。これらと、英語文中にある “Shinjuku” という語との間の類似度を計算し、閾値を越えるものがあれば、“新宿 ↔ Shinjuku” を対応候補とする。

この場合は類似度は 1.0(完全一致) となるが、“ローズワイン ↔ rose wine” の場合は、ローズワインから “rosuwain” という訳語候補が得られ、これと “rose wine” との類似度は 0.78 となる。

これらの語は対訳辞書で対応が見つかることは極めて少ないが、この方法によって非常に高精度に発見することができるようになる。

3.2.5 人称代名詞

日本語文に代名詞の省略があり、ノードが追加されていて、かつ英語文において “I” や “he” などの人称代名詞が対応付けられずに残っている場合、追加ノードと英語の人称代名詞を対応付ける。

3.2.6 統計的手法によるアラインメント

これまでに挙げた方法は、どれもなんらかの知識に基づいたものであり、ロバスト性に欠ける部分がある。つまり特殊な訳語が使われている文や、意訳されている文、専門用語が多く含まれる文などに対しては、上記の方法では対応候補の探索が不十分である場合が多い。また専門用語などは形態素解析が誤る場合もある。

そこで我々は、語の分割位置に依存しないアラインメント手法を提案した Fabien [7] の研究を利用した。Fabien らは日本語のような分かち書き (単語同士の間をスペース区切ること) されていない言語であっても、文字単位でのアラインメントを行うことにより、分割位置に依存しない対訳文アラインメントを可能にした。

この手法では、まず任意の部分文字列の出現頻度と、言語対間の任意の部分文字列同士の共起頻度とを、全対訳文中において計数する。対訳コーパスのサイズが大きくなればなるほど組み合わせは爆発し、メモリを大量に必要とするが、Fabien は Suffix Array[11] を用いることにより効率的に処理している。次にこれらの頻度情報から、任意の部分文字列ペアの相関係数を計算し、相関が強いものを対応とする。

この手法を使う利点は、辞書に載っていようなドメインに特化した訳語を高精度で発見できるとう点である。しかしながらこの手法のみをそのまま利用するだけでは、機能語の扱いが不十分であることや、多くの不適切な対応も発生してしまうなどの問題がある。そこで、我々のアラインメント手法における対訳候補探索手段の一つとすることで、我々のアラインメント精度の向上に期待する。

3.3 対応候補の選択

これまでのステップで、対訳文間の可能な限りの対訳候補が得られた。図 3.1 の例のように、全ての対応候補が完全に 1 対 1 に対応しており、かつ全ての対応が正しく対応付いているならば問題はないが、曖昧性のある対応候補や、文脈上不適切な対応候補が得られる場合もある。図 3.3 の例では、日本語の “保険” と英語の “insurance” がそれぞれ 2 回ずつ出現しているため、組み合わせとして 4 つの対応候補が存在したり、“請求” と “申し立て” にそれぞれ “claim” という訳語があり、対応候補が衝突しているなどの曖昧性がある。このような対応候補の中から、いかに適切な対応を採用するか、またいかに不適切な対応を棄却するかが重要であり、アラインメントの精度、翻訳の精度に大きな影響を与える。

この問題の解決法として、我々は**アラインメントの整合性尺度**を導入することを提案する。これについては、3.6 章で詳しく述べる。

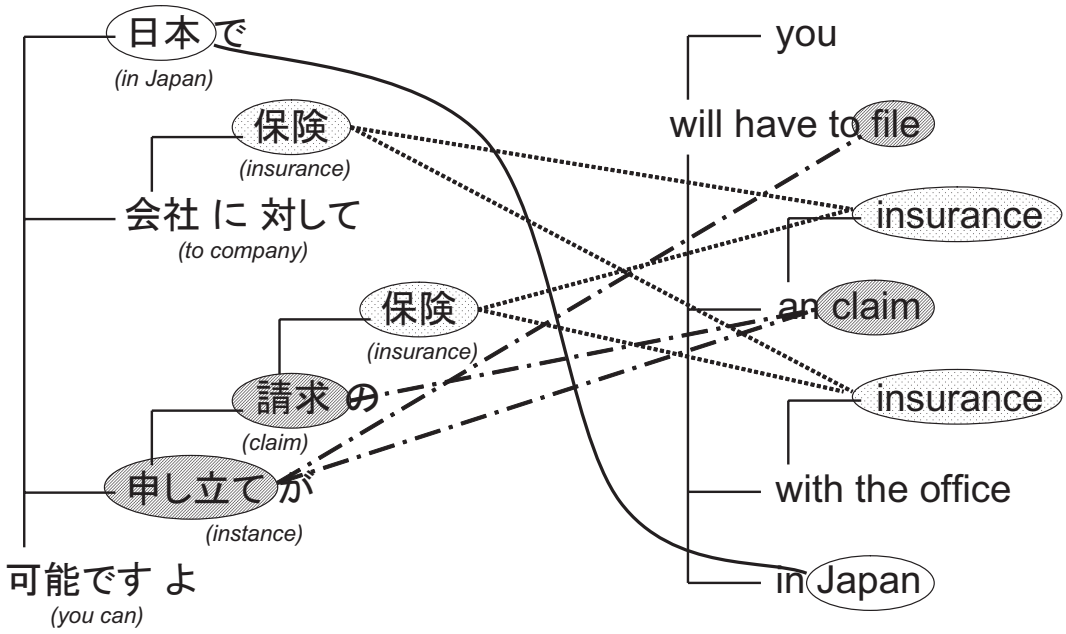


図 3.3: 曖昧性の例

3.4 未対応部分の推定

ここまでの処理で、発見された対応候補から適切なものが取捨選択され、曖昧性のない対応がついているが、対応が全く付かないノードが残る場合がある。ここで対応の付かなかったノードに対する処理を行なう。

まずそれぞれの構文木のルートノードがどちらも対応付いていない場合、それらを対応付ける。次に名詞句内の未対応ノードのうち、同一名詞句内に対応付いたノードがあるならば、未対応ノードを対応付いたノードに併合する。その他の未対応ノードは、そのノードの係り先のノードに併合する。

ただし句読点がある場合や、係り受け情報を見て節の区切りであるとわかった場合には、併合は行わない。このような制限を設けると、最後まで未対応のまま残るノードができる場合がある。両言語において、このような未対応ノードが、すでに対応付いているノードに木構造上で挟まれている場合、未対応ノード同士を対応付ける。

以上の処理を経ても対応付かないノードは、用例としては使わないこととする。

図 3.1 の例では、ルートノード同士はすでに対応付いているので問題ない。次に “あの” が同一名詞句内の “車 ↔ the car” に併合され、最後に “突然”、“at me”、“from the side” がそれぞれの係り先である “飛び出して来たのです ↔ came” に併合される。

ここまでで構築された対応を基本対応と呼ぶことにする。例では以下の 3 つの基本対応が得られることになる。

- “交差 点で ↔ at the intersection”
- “あの 車が ↔ the car”
- “突然 飛び出して来たのです ↔ came at me from the side”

3.5 用例ベースの構築

基本対応がえられたら、各基本対応と、日英両側で連続である (依存構造のどこかで親子関係にある) 基本対応の組合わせすべてを用例ベースに登録する。

図 3.1 の例からは、それぞれの基本対応と、それを組み合わせた “交差点で 突然 飛び出して来たのです ↔ came at me from the side at the intersection” や “突然 あの 車が 飛び出して来たのです ↔ the car came at me from the side” などを用例として登録する。

3.6 整合性尺度を用いた構造的対訳文アラインメント

以上がアラインメントの流れであるが、ここで問題となるのが、3.4 章でも述べたように、曖昧性のある対応候補や、文脈上不適切な対応候補が得られる場合である。

対応候補からの適切な対応の選択を実現し、かつ全体的な整合性を測るために、任意の二つの対応候補 (a_i と a_j とする) の間に**整合性スコア**を定義する。整合性スコアを用いて、以下の式から最も整合的なアラインメントを得る。

$$\underset{alignment}{\text{argmax}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{整合性スコア}\ (a_i, a_j) \tag{3.1}$$

整合性スコアは文の依存構造木上で定義される。

まず、任意の一組の対応候補 a_i(原言語の句 p_{Si} と目的言語の句 p_{Ti} との対応) と a_j (同様に p_{Sj} と p_{Tj}) に注目する。a_i と a_j の原言語側の**距離** $d_S(a_i, a_j)$ を、p_{Si} と p_{Sj} との木構造上の距離と定義する。目的言語側も同様に $d_T(a_i, a_j)$ が定義される。

この距離を用いて、整合性スコアは以下のように定義する：

$$\text{整合性スコア}\ (a_i, a_j) = f(d_S(a_i, a_j), d_T(a_i, a_j))$$

$f(d_S(a_i, a_j), d_T(a_i, a_j))$ は距離のペアに対してスコアを付与する**距離-スコア関数**である。

文全体のアラインメントの整合性は、整合性スコアを用いて式 3.1 のように定義される。また整合性スコアを用いて、各対応候補の整合性も以下のように計算できる：

$$score(a_i) = \sum_{j \neq i}^{n} \text{整合性スコア}\ (a_i, a_j) \tag{3.2}$$

以上の定義により、整合的なアラインメントを得る問題は、**距離**と**距離-スコア関数**をいかに実装するかという問題に置き換わった。

3.6.1　ベースライン手法

ベースライン手法では、曖昧でない対応候補は無条件で採用し、曖昧なものに対してのみ、距離と距離-スコア関数を用いる。

またすべての枝の距離を 1 とする。つまり、ある対応から別の対応への依存構造木上での移動距離をそのまま対応間の距離とする。

また距離-スコア関数は、$f(d_S, d_T) = 1/d_S + 1/d_T$ とする。これは、「曖昧な候補の近くにある (d_S、d_T が小さい) 曖昧でない候補は、曖昧な候補を強く支持する」という仮定に基づいている。図 3.3 にスコア計算の例を示す。

全ての曖昧な候補のスコアを式3.2 に基づいて計算し、最も高いスコアを得たものを採用し、これと衝突する候補を棄却する。曖昧な候補がなくなるまでこの計算を繰り返す。

3.6.2　提案手法

提案手法では**距離**と**距離-スコア関数**を改善し、アラインメントの精度向上を目指す。

距離-スコア関数の学習

距離-スコア関数を改善する。まず毎日新聞 4 万対訳文のアラインメント正解データ [18] から、距離ペアの頻度分布を計数した。

図 3.4 にこの結果を示す。縦軸 (Z 軸) は頻度の log を取ったものである。二つの横軸 (X,Y 軸) は日本語、英語それぞれの距離である。3 次元のプロットを 2 次元で視覚的に捉えやすくするために、別角度からの図を 2 つ示した。分布は、距離が近いペアが最も多く、遠いペアになるに従ってゆるやかに減少し、距離の差が大きくなるに従って急激に減少している。

この結果を元に、次のような基準を考え、距離-スコア関数を人手で設定した：

- d_S・d_T とも小さければ、適切な関係である可能性が高いので、プラスの値を与える
- どちらも大きければ、二つの対応の関係の信頼性が薄いので、スコアは 0 とする
- 一方が小さく、他方が大きければ、不適切な関係である可能性が高いので、マイナスの値を与える

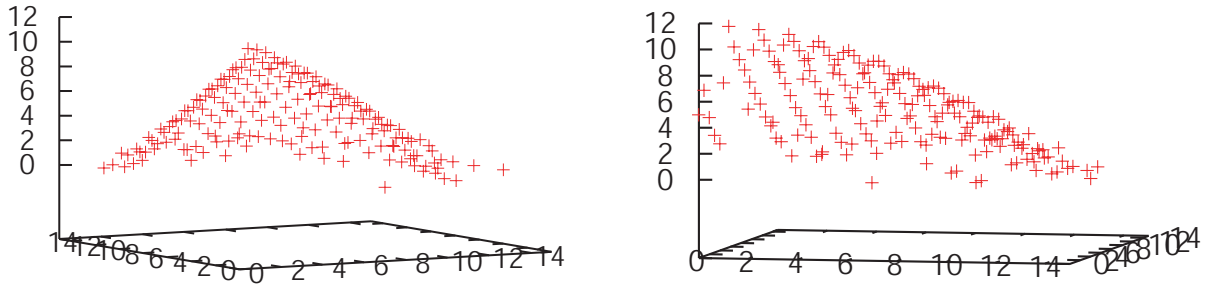


図 3.4: 正解データから学習された距離ペアの分布

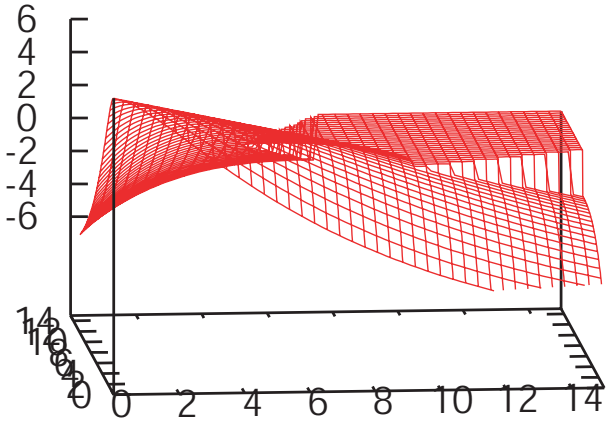


図 3.5: 距離-スコア関数

実際に設定した関数は図 3.5 に示した。

係り受け距離

ベースライン手法では、すべての枝の距離は 1 であるとしたが、実際には文には句や節などの単位が存在し、これらの情報を利用することが有効である。例えば、木構造上では隣接していても、異なる節に属する句同士の距離は大きいと考えられるからである。

日本語構文解析器 KNP が出力する係り受けタイプ情報と、Charniak の nlparser が出力する英語のタグ情報に対して、人手で木構造上での距離を定義した。これを**係り受け距離**と呼び、d_S および d_T の計算に利用する。節などの区切りの強さが強いものほど、距離が大きくなるようにする。図 3.6 にその一部を示す。また、図 3.7、図 3.8 に実際の文での適用例を示す。枝上のラベルが係り受けタイプ、ラベルの上の数字が、係り受け距離である。

図 3.7 の例では、注目する対応同士の距離が、日本語・英語とも 1 であり近いため、適切な関係であると判断し、プラスのスコアを付与する。一方、図 3.8 では、

日本語の係り受け距離

用言:レベル C	6
用言:レベル B+/B	5
用言:レベル B-/A ト格	4
ヲ格/二格/デ格	3
ガ格/ノ格/連体	2
文節内 用言:レベル B+	1

英語の係り受け距離

S/SBAR/SA/:	5
VP/ADVP ADJP/WHADVP WHADJP	4
NP/PP/INTJ QP/PRT/PRN	3
others	2

図 3.6: 係り受け距離

日本語の距離は 1 で近いが、英語の距離は 7 と遠い。このような対応同士は、どちらかが不適切な対応である可能性が高いため、マイナスのスコアを付与する。

またこの距離設定のもとで、前章と同様に距離ペアの頻度分布を係数し、距離-スコア関数を再設定した。

3.6.3 アラインメントの整合性

最良のアラインメントは式 3.1 により、整合性スコアの和が最大になるように、それぞれの対応を採用・棄却していけばよい。しかし全ての場合を調べるのでは候補の数が爆発してしまうので、グリーディーに探索する。

すべての対応の候補について、式 3.2 で表されるスコアを計算する。スコアが最も高いものを採用し、それと衝突する候補は棄却する。同時に、このとき計算されたスコアが閾値を下回る候補があった場合、その候補は誤った対応である可能性が高いため、その場で棄却する。これを繰り返すことにより、近似的に最良のアラインメントが得られる。

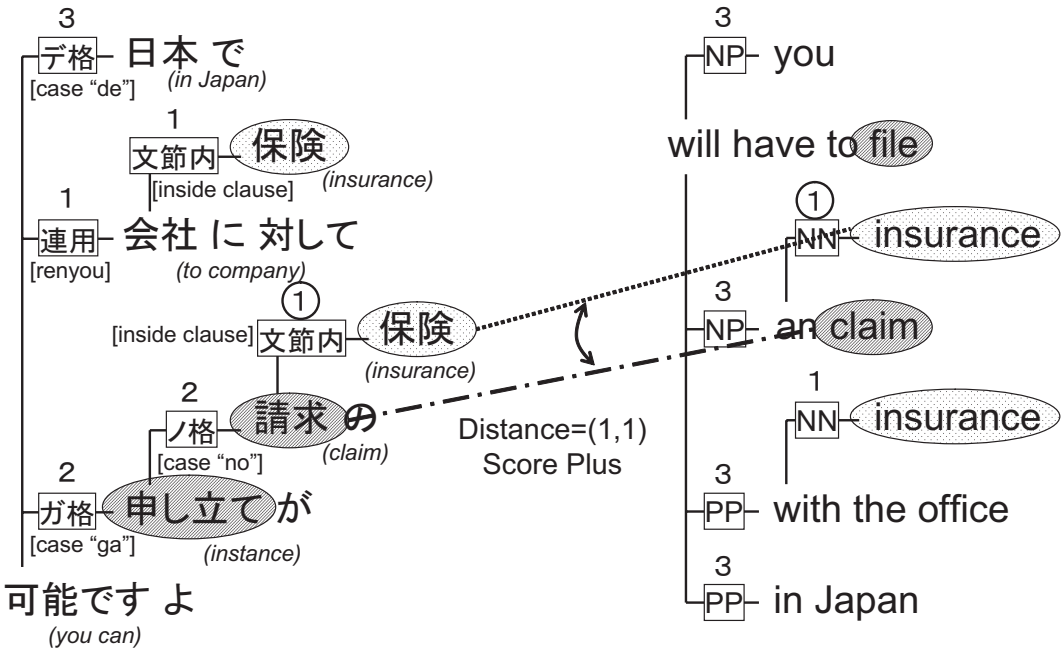


図 3.7: 適切な距離関係の例

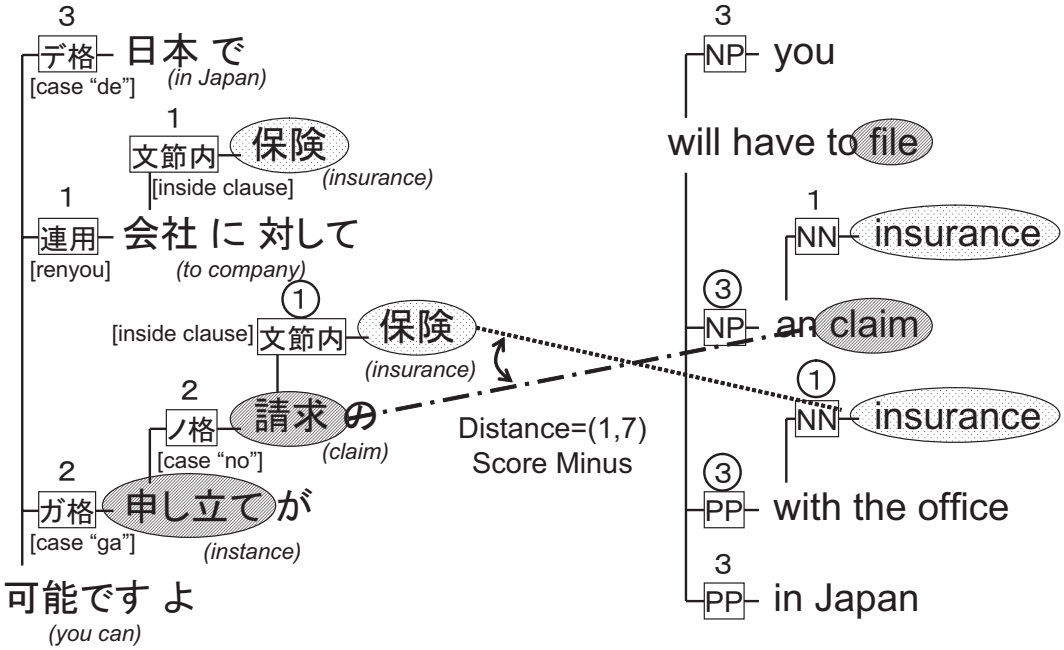


図 3.8: 不適切な距離関係の例

第4章　翻訳

翻訳部では、以下の手順で入力文に対する翻訳を生成する (図 4.1):

1. 日本語入力文を依存構造木に変換
2. 用例の検索
3. 用例の選択
4. 用例の組み合わせ
5. 出力文の整形

4.1　入力文の依存構造解析

アラインメント部と同様に、入力文を依存構造木に変換する。人称推定も同時に行ない、省略された代名詞がある場合には、ノードを追加する。さらに、木構造を SYNGRAPH 化し、柔軟に用例が検索できるようにする。

4.2　用例検索

入力文の各部分木に対して、翻訳に利用できる用例を用例データベースから検索する。翻訳に利用する用例の検索は入力構文木の根 (文のヘッド) から行ない、これを検索対象のルートノードとする。まず検索対象のルートノード単体にマッチする用例を全て検索する。次にルートノードと、木構造上でそれに隣接するノードの組み合わせにマッチする用例を検索する。このように、検索対象のノードを徐々に拡大していき、拡大できなくなったところで終了する。これで、文のヘッドをルートノードとするすべての用例の検索が終了する。

次に、検索対象のルートノードに隣接するノードを新たな検索対象のルートノードとし、再びすべての用例を検索する。これにより、入力構文木の全ての部分木にマッチする用例を検索することができる。

図 4.1 の例では、まず “でした” を根とする部分木 “でした”、“青 でした”、“信号は でした”、“信号は 青 でした” などを順に検索し、次に “青” や “信号は” を根とする部分木の探索を順に行う。

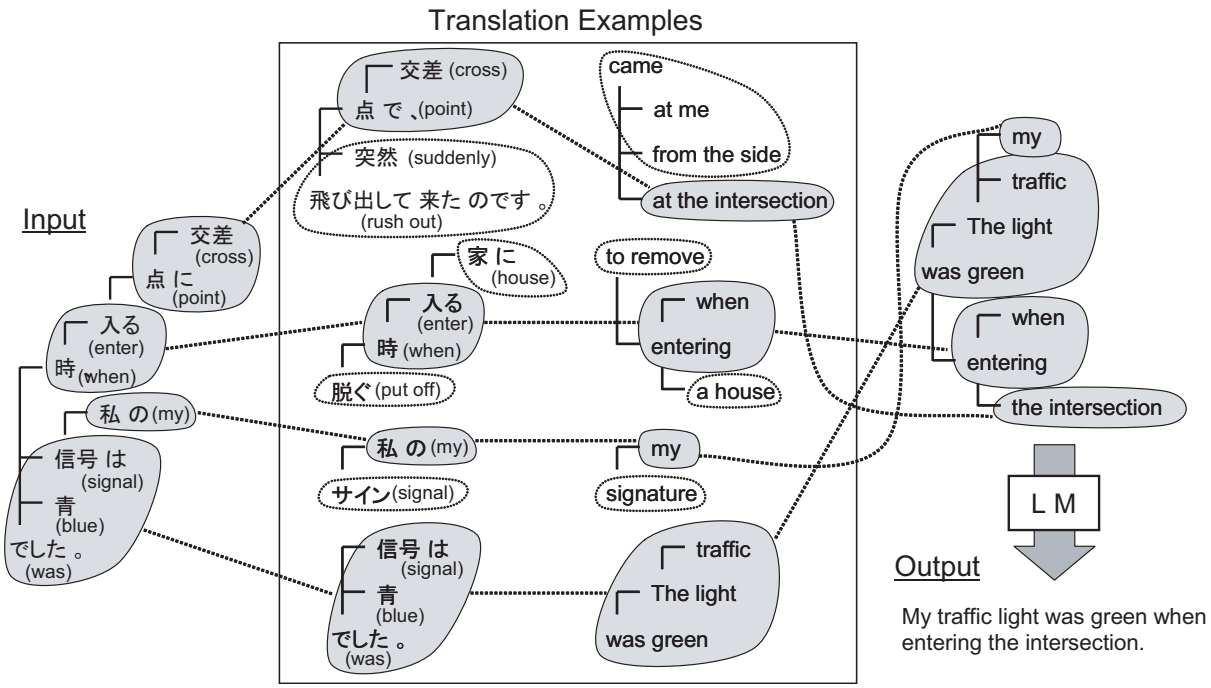


図 4.1: 翻訳の例

入力文のあるノードについて、用例が一つも見つからなかった場合は、対訳辞書を用いて訳を獲得し、これを用例と同様にして扱う。

4.3 用例の選択

次に、検索された用例の中から翻訳に実際に用いるものを選択する。用例の検索時と同様に、文のヘッドから順に注目する。

まず文のヘッドをルートとする全ての用例に対してスコアを付与する。用例ベース翻訳の基本的な考え方として、大きな用例であれば文脈が安定し訳も正確であろうというものがある。このためスコアは入力文とマッチするノード数を基本とする。さらにその外側のノード (親ノードと子ノードのそれぞれ) について、入力文と用例との語の類似度を計算し、類似度に応じて 0〜1 のスコアを与える (類似度が最も高いものが 1)。さらに、子ノードは 2 つ以上存在する場合があるが、入力文の子ノードよりも用例の子ノードが多い場合、多い分だけスコアをマイナスする。

このようにして計算されたスコアを見て、スコアが最も高い用例を翻訳に利用する。このとき、スコアが同じになる用例が複数見つかる場合がある。その場合は、翻訳確率の最も高い用例を利用する。例えば “こんにちは” という日本語に対して、用例の英語が “Hello.” が 3 つ、“Hi.” が 2 つならば、確率の高い “Hello.” を

利用する。

このようにして選択された用例がカバーする日本語のノードに対しては、すでに翻訳が完了しているので、以降は処理は不要となる。次に、カバーされたノードに隣接するノードに注目し、これまでの処理を再び繰り返して用例の選択を行なう。

全ての入力文のノードがカバーされれば、用例の選択は終了である。

4.4 用例の組合わせ

ひとかたまりの用例の内部は、英語の構造、語順などの情報を保持しているため、そのまま利用可能であるが、問題は用例と用例の組合わせ方である。

用例を組合わせるとき、には**糊しろ**情報を利用する。糊しろとは、実際に用例として使うノードの一つ外側にあるノードを指し、この部分に他の用例を張り合わせることができる。糊しろには、子供方向への糊しろと親方向への糊しろの 2 つのタイプがある。

子供方向への糊しろがある場合は、そこに一意に次の用例を貼り付ければよい。たとえば、図 4.1 の例では、“入る時” には “家に” という糊しろがあり、この対応先の英語ノードに “交差点” の対応先英語ノードを貼る、すなわち “a house” に “the intersection” を貼ることにより、二つの用例を組み合わせることができる。

一方、親方向への糊しろの場合は問題が生じる。もし張り合わせる先の親に二つ以上の子ノードがある場合に、他の子ノードの用例との前後関係の情報がないのである。現在のところは、自分よりも前から係る子ノードならば、他の子ノードの前に、自分よりも後ろから係る子ノードならば、他の子ノードの後ろに付加するという単純なルールを用いて組み合わせている。図 4.1 の例では、“私の” には “サイン” という糊しろがあり、“my” は前の子ノードとなることがわかるので、“traffic” よりも前に付加する。

非常にまれではあるが、組み合わせる用例にまったく糊しろがない場合がある。この場合は組み合わせに利用できる情報が全くないので、適当なルールによってコントロールしている。

4.5 出力文の整形

ここまでのステップで、翻訳作業はほぼ終了している。最後に生成された翻訳に対して整形を加える。

4.5.1 数字の翻訳

アラインメントにおいて汎化されていた数字を、適切な形で元に戻す。用例の数字の部分には、英語の数字のタイプが記録されており、これに従って入力文の数字を翻訳する。

例えば “13 日” を翻訳したい場合、用例に “2 日”(実際には “<num:序数> 日” などと汎化されている)がある場合、序数という情報から “13 日 → thirteenth” と翻訳することができる。

4.5.2 言語モデルの利用

人称推定により代名詞は適切に処理されているはずであるが、100%正確というわけではない。翻訳に代名詞がなかったり、二つ重なる場合もある。そこで代名詞の過不足を英語の言語モデルを利用して判別し、修正する。

また冠詞について、“a/an/the” のうちどれを使うか (または不要か) の判断も、言語モデルを用いて行なう。

代名詞、冠詞の選択、有無の全ての可能性を言語モデルにより評価し、最も確率の高い (正確な英語に近い) 文を、最終的な翻訳として出力する。

言語モデルの学習、文の評価には CMU の Cam_Toolkit[6] を用いた。

第5章　解析結果表示ツール

5.1　背景

これまでの自然言語処理 (NLP) 研究では、マシンパワーや利用可能なリソースなどの問題で、研究の対象がそれほど大規模ではなく、また処理の複雑さも問題視されるようなことはなかった。しかし近年の NLP 技術の目覚しい発展や、マシンパワーの向上、さらには Web などに代表されるような大規模リソースへの容易なアクセスが可能となったことにより、研究対象がより複雑化、大規模化している。必然的に、それらを正確に扱う処理 (プログラム) も複雑化、大規模化している。

研究においては、あるシステムで実験を行い、得られた実験結果を議論し、システムへのフィードバックを与えるというプロセスが必須であるが、システムの複雑化により、実験結果を議論する際の手間が無視できないぐらいに増えてきている。つまり、解析ミスなどの言語現象の原因が、処理の複雑化により、複合的なものになる場合や、リソースの大規模化により、そのリソース内の部分を検索するのにも時間がかかる場合がある。

このように我々は、作業や研究の効率化のために、システムの解析結果を視覚的・直感的にわかりやすい形で表示できるようなツールの必要性を感じ、その構築を試みた。

この際、ツールを深く作りこんでしまい、あるシステムに特化したようなものにしてしまうと、外のシステムへの流用が困難となり、その都度ツールを作りなおさねばならず、本末転倒になってしまう。そこでツール自体には必要最低限、かつ十分な機能のみを搭載したシンプルなものにとどめておき、柔軟性かつ頑健性を持たせることを考えた。

さらに、誰もが容易にアクセスでき、多人数での議論においても有用なものにするため、CGI で実現し、Web ブラウザでアクセスすることを考えた。Web ブラウザはどんな種類のマシンにも搭載されており、プラットフォームを気にする必要がないため、十分な汎用性・簡便性を確保できる。また、HTML の機能を用いることにより、視覚的にもわかりやすい表示を実現可能である。

Web 上で視覚的なわかりやすさ、簡便さを得るために、我々は HTML タグであるテーブル (表) を利用することを考えた。どのテーブルのどのセルに何を表示するのかを指示することによって、様々な種類の解析結果を表示することができる。このとき、表示ツール自体は、指示された内容を表示するだけであることが重要であり、これによって、表示させたい内容によってツールを作り替える必要は一

切不要である。表示内容は各システムが、それぞれ知りたい情報を出力すればよいのである。

5.2 スペック

表示ツール自体は、各セルごとに指定された内容を持つ、複数のテーブルを表示する機能しか備えていない。各セルに表示する内容やスタイルは、アプリケーション側が指定することになる。

以下に挙げる仕様に従って記述された表示指定ファイルを用意し、CGI に渡すと、指示通りのテーブルが表示される。

5.2.1 セルの指定

セルは、3つの番号のペアで指定する。1つ目はテーブルの番号、2つ目は行番号 (縦位置)、3つ目は列番号 (横位置) である。命令行であることを示す%%の後にこの3つの数字を置いて、セルの指定をする。例えば、

```
%% 1 2 3
ここにセルに表示したい内容を
書きます。
```

という指定は、1つ目のテーブルの2行3列のセルに、“ここにセルに表示したい内容を書きます。” と書く。命令行から次の命令行までの内容が、セルの内容となる。ただし、行間にある改行は無視されるため、改行したいときは明示的に
を挿入する必要がある。

また、セル内の文字の位置や、セルの背景色などを指定したい場合は、

```
%% 1 2 3 valign=top bgcolor=red
この後に
セルに表示したい内容を
書きます。
```

のように、セル指定数字の後にスペース区切りで、プロパティと値を= でつないで指定することができる。これらのプロパティは HTML に準拠する。上の例では、縦位置を上詰めにして、背景を赤にする。

5.2.2 テーブルフィーチャ

テーブルにフィーチャを指定することもできる。

```
%% 2 width=80%
```

この指定では、2つ目のテーブルの幅が、ブラウザの幅の 80%となる。

5.2.3　ページフィーチャ

ページタイトルなどのページ全体のフィーチャは、%%の直後に指定する (数字をつけない)。

```
%% title=表示ツール
```

この指定では、ページのタイトル (<HEAD>内の<TITLE>タグに相当) を “表示ツール” にする。

5.2.4　追加情報の表示

テーブル内に表示すると繁雑になりすぎてしまう情報は、リンクとして別のウインドウで表示させることができる。リンク先に表示させる内容は別ファイルで用意する必要はなく、同じファイル内に埋め込むことができる。

リンク元のセルのフィーチャに label=LABEL1 と、ラベルフィーチャを指定し、以下のように内容を記述する。

```
%% label=LABEL1
リンク先に表示したい内容
```

5.3　使用例

表示ツールを使って、「井川投手は来年１月上旬に渡米し、ニューヨークで入団発表の記者会見に臨む予定。」という文を構文解析した結果を表示した例を図 5.1 に示す。図 5.1 の表の右側の情報はリンクになっており、クリックすることで別ウインドウでさらに詳しい情報を見ることができる (図 5.2)。

機械翻訳での使用例は、7 章で示す。

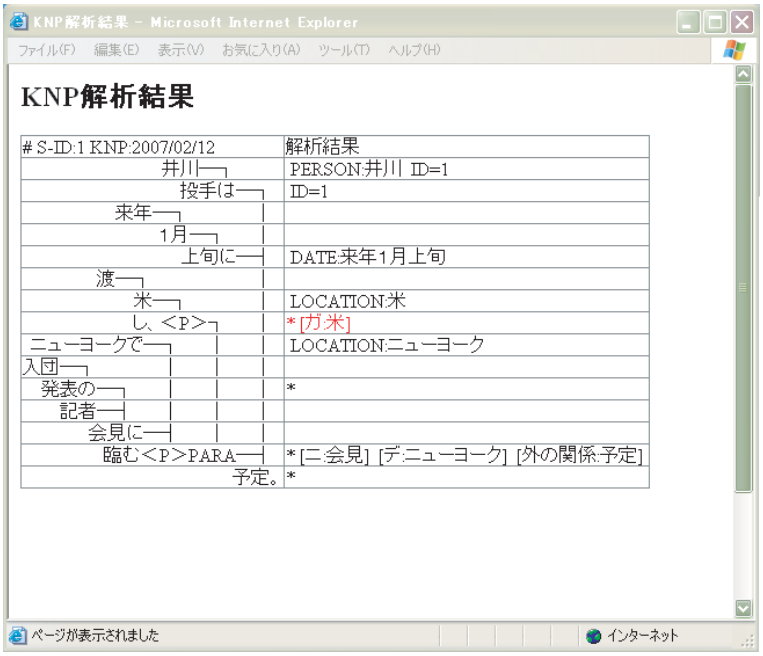


図 5.1: KNP の解析結果表示例

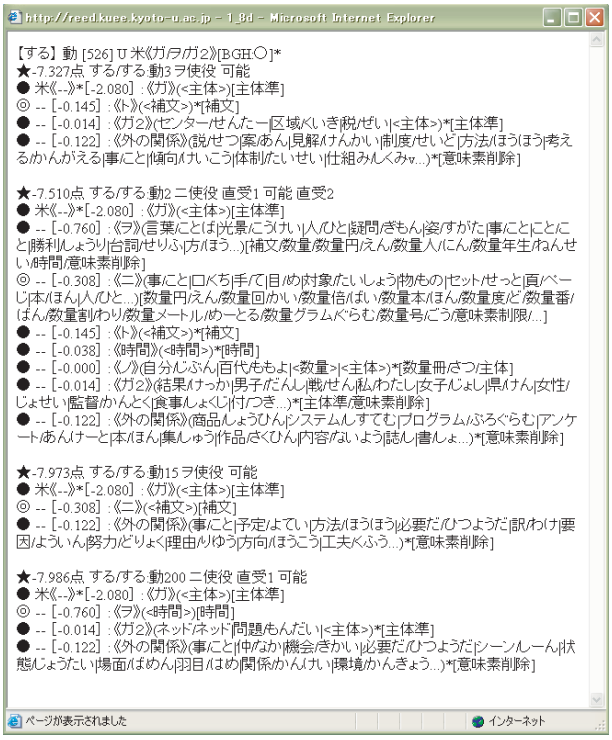


図 5.2: 追加情報の表示例

第6章　関連研究

6.1　Logical Form

先に挙げた用例ベース翻訳の例では、対訳文の形態素解析/構文解析などを行ない、依存構造に変換してから、対訳文内のアラインメントを取っており、入力文の構造そのままの形を用いてる。

これに対して Arul ら [12] は、入力文を Logical Form (LF) という形に変換してから、同様に対訳文内のアラインメントを取っている。LF は、文の中でもっとも重要な意味を持ついくつかの要素 (内容語) 同士の関係を、順序なしのグラフを用いて表現したものである。各ノードは内容語の原形であり、枝はそれぞれのノード間の意味関係を表わす。

LF では、語順、活用、機能語などの各言語に特徴づけられる要素を一切排除しており、これにより非言語依存で頑健な翻訳システムの構築を可能としている。図 6.1 にスペイン語と英語における LF、およびそのアラインメントの例を示す。英語側の元の文は “Under Hyperlink Information, click the hyperlink address” である。なおアラインメントの取り方、及び翻訳時の用例の選択方法は我々が開発している用例ベース翻訳システムのそれと大差はないため、ここでの説明は省略する。

6.2　用例検索の効率化

用例ベース翻訳においては、入力文やその各部分に対して利用可能な用例を、大量の対訳コーパスから探す操作が必須である。単純に全ての候補文を走査するのでは、対訳コーパスのサイズに比例して用例探索に時間がかかることになる。この問題を解決を試みた研究に、土居ら [24] の研究がある。

この研究は、隅田 [16] が提案した用例ベース翻訳システム (D^3 と呼ばれる) における用例検索の効率化を図ったものである。D^3 では、入力文と用例候補文との単語レベルでの編集距離を計算し、それが閾値以下である候補文をすべて調べ、そこから最適な用例を検索するのだが、当然すべての候補文について編集距離を計算するのでは、時間がかかりすぎてしまう。

そこで土居らの研究では、候補文集合の分割、単語グラフ、A^* アルゴリズムを利用して効率的な検索を実現している。

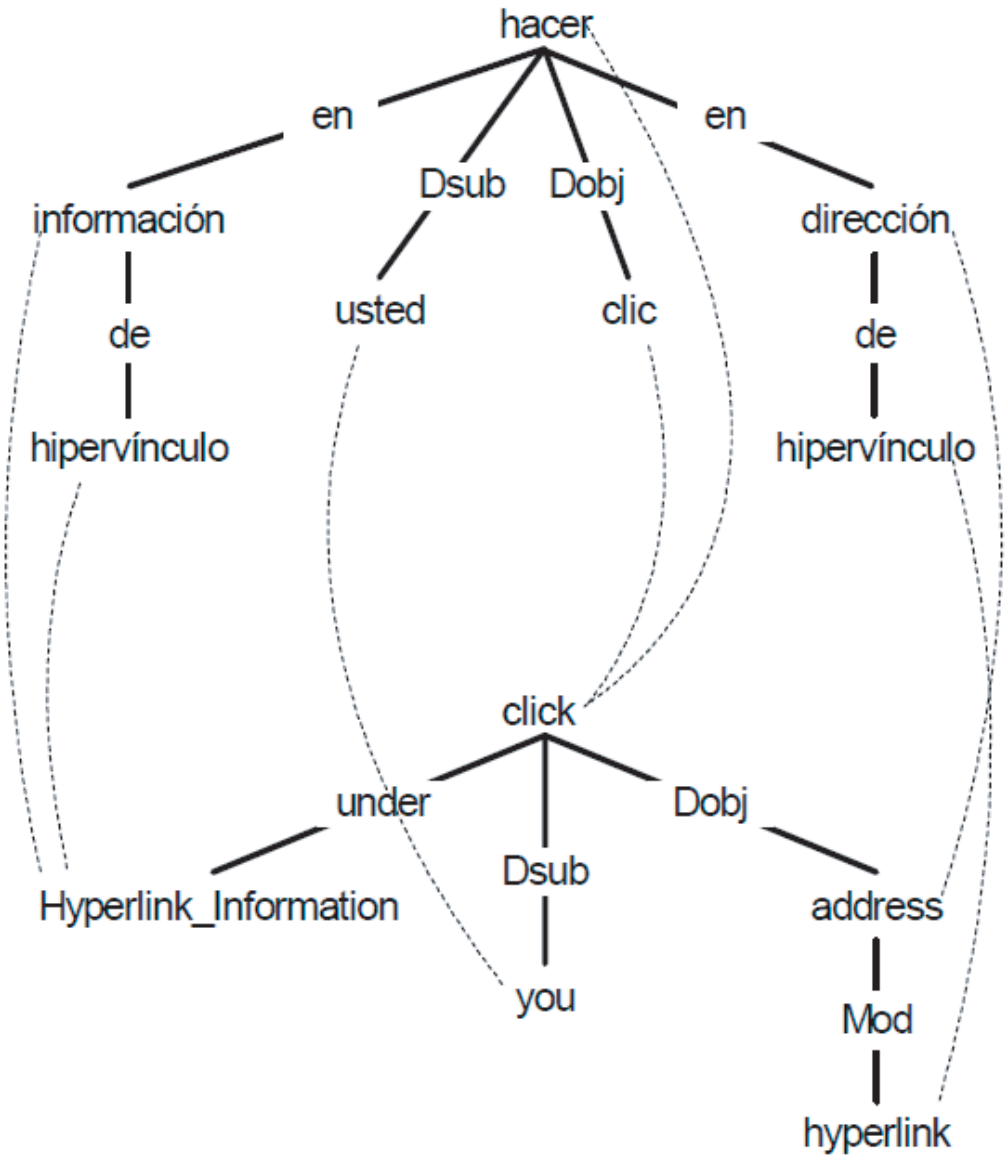


図 6.1: Logical Form を用いたスペイン語-英語アラインメントの例

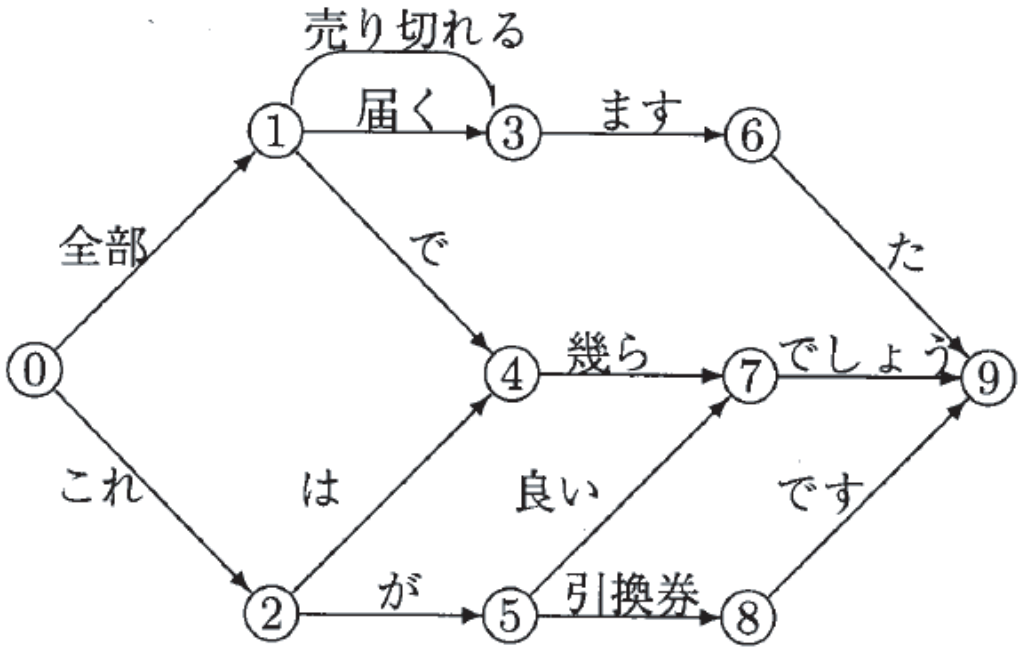


図 6.2: 単語グラフの例

6.2.1 候補文集合の分割

まず内容語数と機能語数によって、すべての候補文をグループ分けする。入力文の内容語・機能語と、各グループの内容語・機能語は完全に一致するものと考え、それぞれの語数から各グループごとに可能な最小距離を求め、この最小距離が閾値の範囲内で小さいグループから順に、検索を進めるのである。

あるグループから最適な候補文が見つかったなら、その距離を新たな閾値とすることにより、検索対象のグループはさらに絞られることになる。

6.2.2 単語グラフ

各グループに属するすべての候補文は、図 6.2 のような単語グラフにまとめられる。単語グラフは有向グラフであり、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いに対応する。この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。

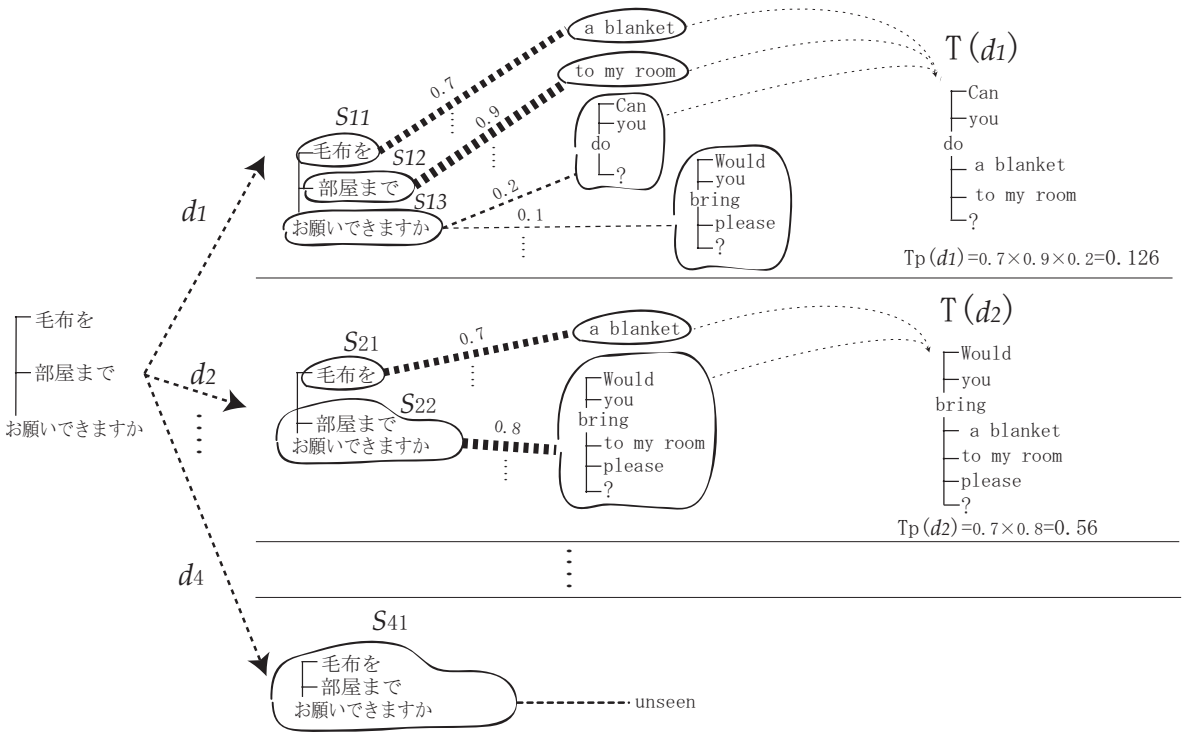


図 6.3: 用例ベース翻訳の確率的定式化

6.2.3 A* アルゴリズム

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。この探索問題を解くために、A* アルゴリズムを用いる。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。侯補文探索問題においては、状態は、単語グラフの経路と入力文との編集距離計算の途中経過を意味する。

6.3 用例ベース翻訳の確率的定式化

用例ベース翻訳は、用例の大きさ、類似度などを経験則による指標で計算してきたため、統計翻訳に比べてアルゴリムが不透明でアドホックであるという問題があった。このような問題を解決するために、荒牧ら [1] は用例ベース翻訳の確率的定式化を行った。ここではこの手法について説明する。

まず、入力文の可能な部分木の組合せを考える。

$$D = \{d_1, ..., d_N\}. \tag{6.1}$$

ここで、d_i は入力文の分解のパターン、D は d_i の集合とする。例えば、図 6.3 の入力文の場合、$d_1, ..., d_4$ の 4 通りの部分木の組合せが可能である。

次に、d_i は入力文を M_i 個の部分木に分解しているとする。

$$d_i = \{s_{i1}, s_{i2}, ..., s_{iM_i}\}. \tag{6.2}$$

s_{ij} は入力文の部分木である。例えば、図 6.3 では、d_1 は入力文を 3 つの部分木 s_{11}, s_{12}, s_{13} に分解している。

ここで、各部分木 s_{ij} について、翻訳確率 $P(t_{ij} \mid s_{ij})$ のもっとも高い用例を選ぶ。これは統計翻訳の場合と同様に、アラインメントされた対訳データから計算することができる。そして、それらの確率の積を分解 d_i に対する翻訳確率 $P(d_i)$ とする。

$$P(d_i) = \prod_{s_{ij} \in d_i} \max_{t_{ij}} P(t_{ij} \mid s_{ij}). \tag{6.3}$$

ここで、d_i の翻訳は $t_{i1}, ..., t_{iMi}$ であり、これを $T(d_i)$ と表記する。

最後に、もっとも高い翻訳確率を持つ d_m を選択する。

$$d_m = \arg\max_{d_i \in D} P(d_i). \tag{6.4}$$

そして、これに対応する翻訳 $T(d_m)$ を最終的な翻訳結果とする。

例えば、図 6.3 の $T(d_1)$ のように、入力文を小さな部分木に分解した場合は、“お願いできますか” に対して様々な英語表現が考えられる。この場合、適切な訳である P(Would you bring | お願いできますか) の翻訳確率は必ずしも高くなく、適切な翻訳が行われない場合もある。

一方、$T(d_2)$ では、より大きな用例 “部屋までお願いできますか” を用いている。この用例の英語表現としては、多くが “Would you bring ... to my room?” となり、この翻訳確率は高い値となる。その結果、用例全体の翻訳確率の積である $P(d_2)$ も高くなり、$T(d_2)$ が翻訳として採用される。

先に述べたように、統計翻訳の研究においても単語より大きい単位を考えたり、構文情報を利用する方向に進んでおり、ここに示したように用例ベース翻訳についても確率的な定式化を考えることが可能である。統計翻訳と用例ベース翻訳は今後さらに近づいていくのではないかと考えられる。

6.4 機械翻訳の評価

科学技術全般において評価は非常に重要であり、客観的な評価尺度が設定できれば、技術の進展を促す効果も大きい。しかし、翻訳は高度に知的な処理であり、その結果を客観的、自動的に評価することは非常に難しいと考えられてきた。

この問題に対して、近年、BLEU[15]、NIST[8] などの自動評価尺度が提案された。これらの基本的な考え方は、表 6.1 に示すように、正解翻訳を複数用意してお

表 6.1: 自動評価のための正解翻訳

入力	毛布を部屋までお願いできますか。
翻訳 1	Would you bring a blanket to my room , please?
翻訳 2	Would you mind bringing a blanket to my room?
翻訳 3	Please bring a blanket to my room.
翻訳 4	Can I have a blanket brought to my room?

表 6.2: 自動評価尺度

BLEU	正解との n-gram の適合率の相乗 (幾何) 平均
NIST	正解との n-gram の適合率の相加 (算術) 平均
WER	Word Error Rate. 正解との編集距離
PER	Position Independent Word Error Rate. 語順を用いない正解との編集距離
GTM	General Text Matcher. 正解との一致した最長語列の適合率、再現率の調和平均

き、それらとの「近さ」を評価するというものである。たとえば、BLUE や NIST では、システムの翻訳結果の n-gram（1 単語、2 単語、3 単語などの部分単語列）が正解翻訳に含まれる割合を基準としている。表 6.2 に代表的な自動評価尺度を示す。

このような自動尺度はもちろん万能ではないが、人間による主観評価と高い相関があり、機械翻訳の質が非常に高い場合は別として、現状の翻訳システムの評価としては一定の役割を果たすと考えられている。

第7章　実験と考察

7.1　アラインメント精度

毎日新聞対訳コーパス [18] からランダムに 500 文を選び、アラインメントを行なった。このコーパスにはアラインメントの正解データが付与されている。

また対訳辞書として、研究社の日英辞書 (36K 見出しから 214K 対訳を抽出) と英日辞書 (50K 見出しから 303K 対訳を抽出) を利用した。

評価の単位は、英語は単語単位、日本語は文字単位とした。これは、正解データも我々の出力も句単位なのだが、日本語の句の区切りが必ずしも一致しないためである。評価は、各文ごとに正解データとの適合率・再現率・F 値を算出し、その平均値を計算した。

図 7.1 を用いて適合率の計算例を示す。太い罫線で囲まれた部分が正解のアラインメントであり、黒い四角が出力であるとする。適合率は、出力のうちの正しいものの割合を表しており、例では出力 12 箇所のうち 9 箇所が正しいので、適合率は 9/12 で 75%となる。

同様に再現率は正解をどれだけカバーできたかを示す値であり、図 7.2 の例では、正解 11 箇所のうち 9 箇所をカバーできているので、再現率は 9/11 で約 82%となる。

F 値は適合率を P、再現率を R として、以下の式で計算される:

$$\text{F 値} = \frac{2PR}{P+R} = \frac{2 * 0.750 * 0.818}{0.750 + 0.818} = \text{約 } 78\%$$

実験結果を表 7.1 に示す。+距離-スコア関数学習はベースライン手法の距離-スコア関数を改善し、学習して得られたものに変えた結果で、+係り受け距離はさらに係り受け距離を利用した結果である。

距離-スコア関数を改善することにより再現率は若干下がったものの、適合率が大幅に向上し、F 値では 2.1 ポイントの精度向上が見られた。また係り受け距離を利用することにより、さらに精度が向上し、ベースラインよりも 3.1 ポイント以上の精度向上を達成した。

この結果から、我々の提案する**距離-スコア関数**と**係り受け距離**がアラインメントに効果的に働いていることがわかる。

実際にアラインメントが改善された例を図 7.3 と図 7.4 に示す。なおこのアラインメントの表示には 5 章で述べた表示ツールを利用している。このツールを利

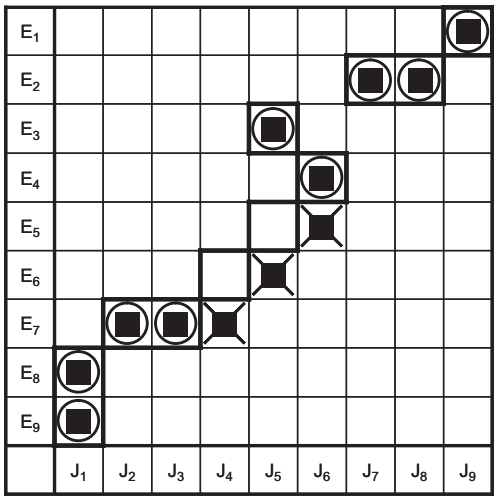


図 7.1: 適合率の計算例

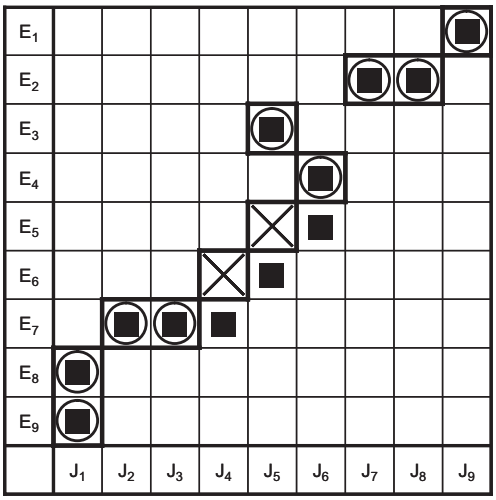


図 7.2: 再現率の計算例

表 7.1: アラインメント精度の実験結果

	適合率	再現率	F 値
ベースライン	60.26	61.68	58.79
+距離-スコア関数学習	64.35	61.58	60.81
+係り受け距離	**64.93**	**62.64**	**61.91**

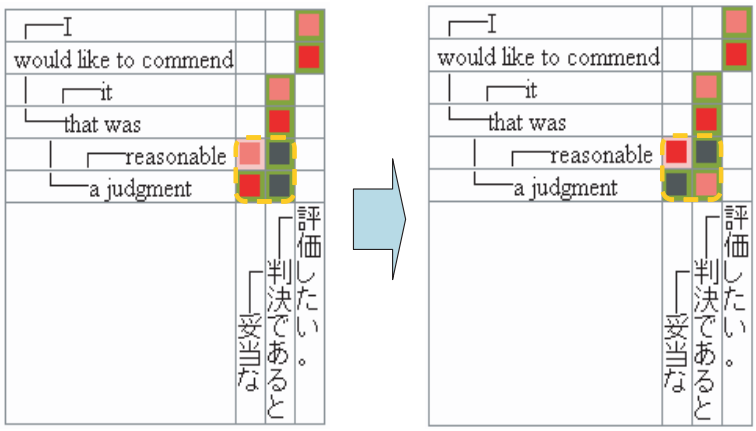
用することにより、対応がどの情報から得られたものか (対訳辞書/柔軟マッチング/Transliteration など) や、最終的に棄却された対応と採用された対応の区別などが、視覚的にわかりやすく表示されていることがわかる。 なお例においては、濃い赤が採用された対応、薄い赤が併合された対応、黒は棄却された対応である。

改善例の黄色い破線で囲まれた部分に注目すると、ベースライン手法で誤って対応付けられたり、正しい対応が棄却されたりしたものが、提案手法によって改善され、適切に対応付けられていることがわかる。

この実験により、我々の提案するアラインメント手法が有効であると言える。

7.2 翻訳精度

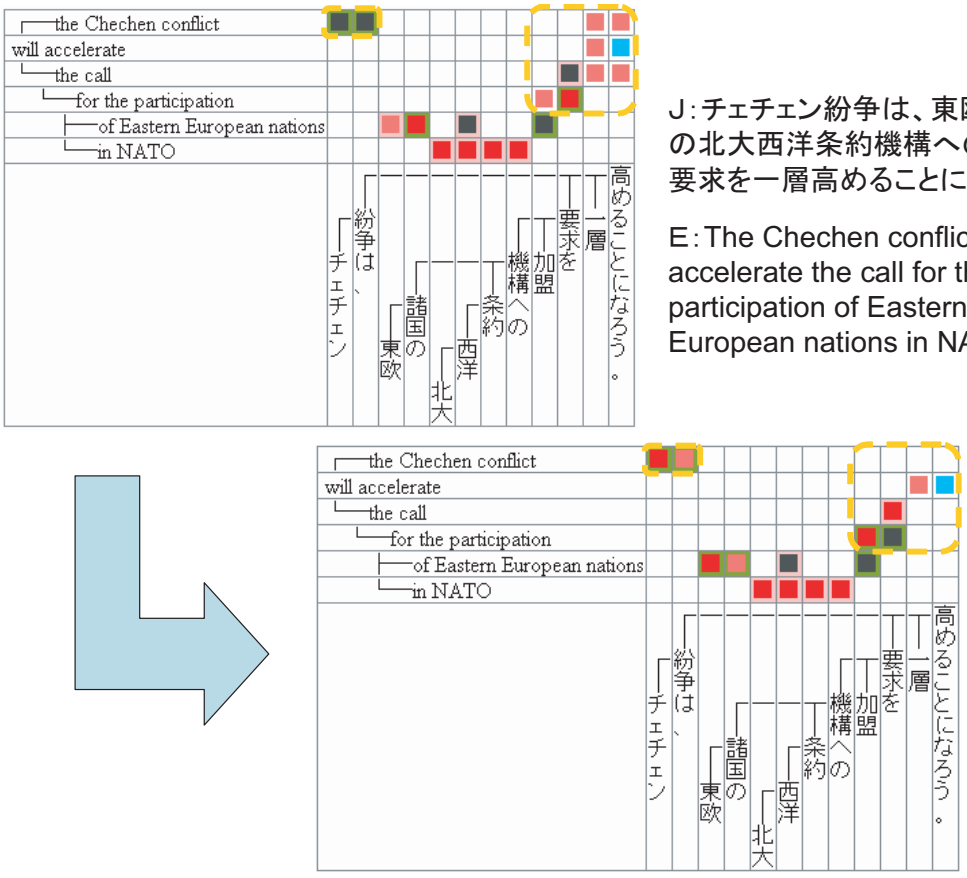
アラインメントのベースライン手法と提案手法とで、翻訳精度の比較実験を行なった。この実験には、BTEC という旅行対話コーパスを利用した。トレーニングデータ (用例を学習するデータ) として 4 万対訳文あり、500 文のテストセットを翻訳した。なお提案手法は距離-スコア関数と係り受け距離のどちらも改善したものを利用した。



J:妥当な判決であると評価したい。

E:I would like to commend that it was a reasonable judgment.

図 7.3: 改善例 1



J:チェチェン紛争は、東欧諸国の北大西洋条約機構への加盟要求を一層高めることになろう。

E:The Chechen conflict will accelerate the call for the participation of Eastern European nations in NATO

図 7.4: 改善例 2

表 7.2: 翻訳精度の実験結果

	BLEU	NIST
ベースライン	11.91	44.74
提案手法	12.05	45.05

結果を表 7.2 に示す。わずかながら精度が向上しており、提案手法によってアラインメントが改善され、用例の質が向上したため、結果的に翻訳の精度も向上したものと考えられる。

第8章　おわりに

本論文では我々の構築した用例ベース機械翻訳システムについて説明し、そこで利用されている高度な言語処理技術として日本語の柔軟マッチングと人称推定について述べた。

さらにアラインメントにおいて、距離-スコア関数 $f(d_S, d_T)$ と係り受け距離を利用した新しいアラインメント手法を提案した。アラインメント全体の整合性を全ての対応候補のペアのスコアの和で定義し、適切な候補の選択を可能にした。これにより、3ポイント以上のアラインメント精度の向上を達成した。

また翻訳実験においては、高精度の翻訳を実現することができた。

今後の課題は、人手で設定している係り受け距離を自動学習で獲得することである。これには、各言語で独立に学習する方法や、言語ペアで学習する方法などいくつか考えられるが、その中から最適な方法で学習することを考えている。

さらに現在我々のシステムは日英翻訳を対象としているが、他言語にも対応可能なシステムにする必要がある。特に、日中翻訳を実現することを考えている。

謝辞

本研究に対して、さまざまなご意見・ご指導をいただき、また、多方面からのご協力を賜わりました、江崎浩教授、黒橋禎夫教授に心より感謝いたします。

また独立行政法人情報通信機構の河原大輔氏にも、様々なご協力をいただいたので、この場を借りて御礼を申し上げます。

さらに日ごろからお世話になっております研究室の皆さまにも心より感謝いたします。

参考文献

[1] Eiji Aramaki, Sadao Kurohashi, Hideki Kashioka, and Hideki Tanaka. Probabilistic model for example-based machine translation. In *Proceedings of MT Summit X*, pp. 219–226, 2005.

[2] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Association for Computational Linguistics*, Vol. 19, No. 2, pp. 263–312, 1993.

[3] Michael Carl and Andy Way. *Recent Advances in Example-based Machine Translation*. Kluwer Academic Publishers, 2003.

[4] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 132–139, 2000.

[5] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 173–180, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[6] Philip Clarkson and Ronald Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of the European Conference on Speech Communication and Technology*, pp. 2707–2710, 1997.

[7] Fabien Cromieres. Sub-sentential alignment using substring co-occurrence counts. In *ACL*. The Association for Computer Linguistics, 2006.

[8] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *In Proc. ARPA Workshop on Human Language Technology*, 2002.

[9] Sadao Kurohashi and Makoto Nagao. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, Vol. 20, No. 4, pp. 507–534, 1994.

[10] Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pp. 22–28, 1994.

[11] U. Manber and G. Myers. Suffix arrays: A new method for on-line string searches. *siamjour*, Vol. 25, No. 5, pp. 935–948, 1993.

[12] Arul Menezes and Stephen D. Richardson. A best-first alignment algorithm for extraction of transfer mappings from bilingual corpora. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL) Workshop on Data-Driven Machine Translation*, pp. 39–46, 2001.

[13] Makoto Nagao. A framework of a mechanical translation between Japanese and English by analogy principle. In *International NATO Symposium on Artificial & Human Intelligence*, 10 1981.

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

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pp. 311–318, 2002.

[16] Eiichiro Sumita. Example-based machine translation using dp-matching between word sequences. In *Proc. 39th ACL workshop on DDMT*, pp. 1–8, 2001.

[17] Eiichiro Sumita, Hitoshi Iida, and Hideo Kohyama. Translating with examples: A new approach to machine translation. In *Proceedings of the 3rd TMI*, pp. 203–212, 1990.

[18] Kiyotaka Uchimoto, Yujie Zhang, Kiyoshi Sudo, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara. Multilingual Aligned Parallel Treebank Corpus Reflecting Contextual Information and Its Applications. In *Proceedings of the MLR2004: PostCOLING Workshop on Multilingual Linguistic Resources*, pp. 63–70, 2004.

[19] Warren Weaver. *Letter to Norbert Wiener*. Rockefeller Foundation Archives, 1947.

[20] 岡嶋穣. モダリティーを用いた省略格の人称推定と日英アラインメントの高度化, 2005. 東京大学卒業論文.

[21] 佐藤理史. MBT1: 実例に基づく訳語選択. 人工知能学会誌, Vol. 6, No. 4, pp. 128–136, 1991.

[22] 佐藤理史. MBT2: 実例に基づく翻訳における複数翻訳例の組合せ利用. 人工知能学会誌, Vol. 6, No. 6, pp. 75–85, 1991.

[23] 大西貴士, 黒橋禎夫. 国語辞典からの類義表現抽出と SYNGRAPH データ構造による柔軟マッチング. 言語処理学会第 12 回年次大会, pp. 1127–1130, 2006.

[24] 土居誉生, 隅田英一郎, 山本博史. 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. 情報処理学会, Vol. 45, No. 6, 2004.