輪講発表資料 平成 17 年 10 月 21 日

構造的言語処理を指向する用例ベース翻訳
Example-based Machine Translation Pursuing Fully Structural NLP

黒橋研究室 電子情報学専攻 56430 中澤 敏明

Abstract

Machine Translation(MT) has been studied for a long time since 1940s. However, high-accurate MT system hasn't been developed yet because MT sysytem requires great power of computers and language resources/knowledge such as large parallel corpus and high-level Natural Language Processing (NLP) technologies. These problems are gradually solved because of the recent remarkable improvement of computers, availability of large parallel corpus through the Internet, and development of NLP technologies. We now have more opportunities to face foreign languages, so the demand for MT is increasing.

In this paper, we briefly show the history of MT first, then explain the basic thoughts of representative methods for MT, Statistical Machine Translation (SMT) and Example-based Machine Translation (EBMT). After that, we introduce some related work for EBMT. Finally, we describe our EBMT system.

1 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。

本輪講では、機械翻訳の歴史を簡単に述べ、機械翻訳の代表的な手法である「統計翻訳 (Statistical Machine Translation)」と「用例ベース翻訳 (Example-based Machine Translation)」の基本的な考えを説明した後、用例ベース翻訳に関するいくつかの研究を紹介し、最後に我々が研究している用例ベース翻訳システムについて述べる。

2 機械翻訳の歴史

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 [11] 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。

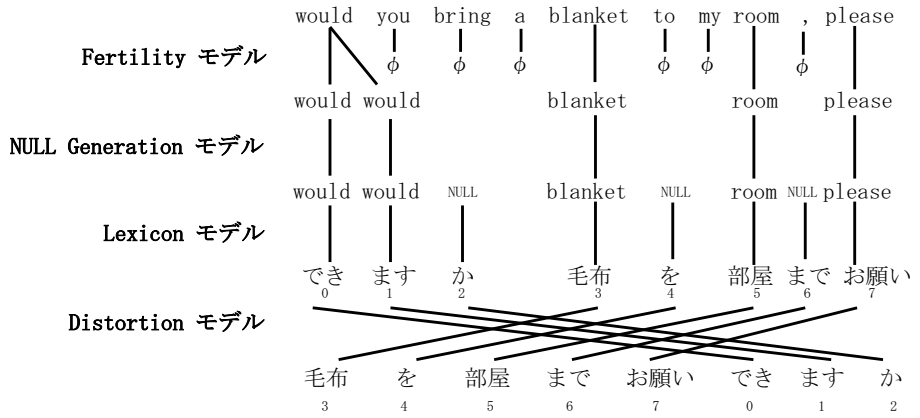

Figure 1: Translation models.

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 [9] 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「統計ベース翻訳」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。現在ではこの統計ベース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。

3 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。

3.1 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。

3.2 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。

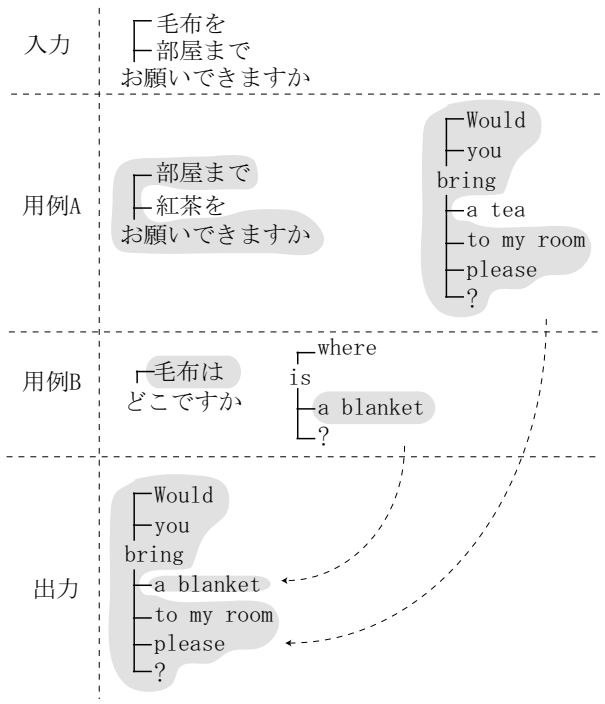

Figure 2: Flow of EBMT.

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 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。

4 用例ベース翻訳

用例ベース翻訳では、利用する言語資源に制限をつけることはせず、対訳辞書や言語知識があるならば、それらを利用して翻訳を行なう。

4.1 用例ベース翻訳の流れ

小さな単位を取り扱う統計翻訳に対し、用例ベース翻訳では、なるべく大きな単位を扱うことを前提としている。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ない、それらを組み合わせることによって翻訳文を生成する。

たとえば、図 2 の例では、入力文中の「部屋までお願いできますか」は用例Aの一部の翻訳を、「毛布」は用例Bの一部の翻訳を利用し、これらの組み合わせによって翻訳文を作りだしている。

また図 2 を見ると、対訳文は形態素解析/構文解析などの処理がされており、依存構造の形に変換されていることがわかる。

4.2 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。

$$D = \{d_1, ..., d_N\}. \tag{1}$$

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 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。

$$d_i = \{s_{i1}, s_{i2}, ..., s_{iM_i}\}. \tag{2}$$

s_{ij} は入力文の部分木である。例えば、図 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。

$$P(d_i) = \prod_{s_{ij} \in d_i} \max_{t_{ij}} P(t_{ij} \mid s_{ij}). \tag{3}$$

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。

$$d_m = \arg\max_{d_i \in D} P(d_i). \tag{4}$$

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。

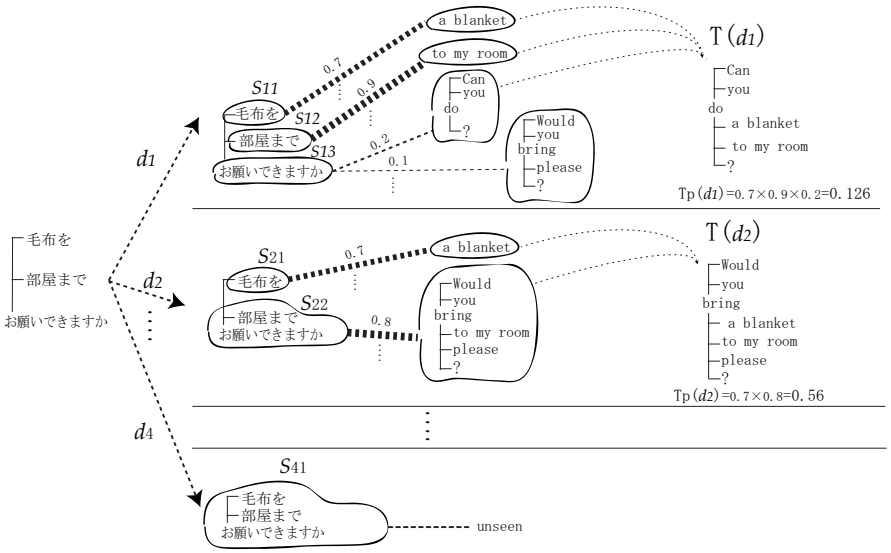

Figure 3: Probabilistic model for EBMT.

例えば、図 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。

4.3 用例ベース翻訳の利点

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。

5 関連研究

5.1 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 [8] 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 4 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。

5.2 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

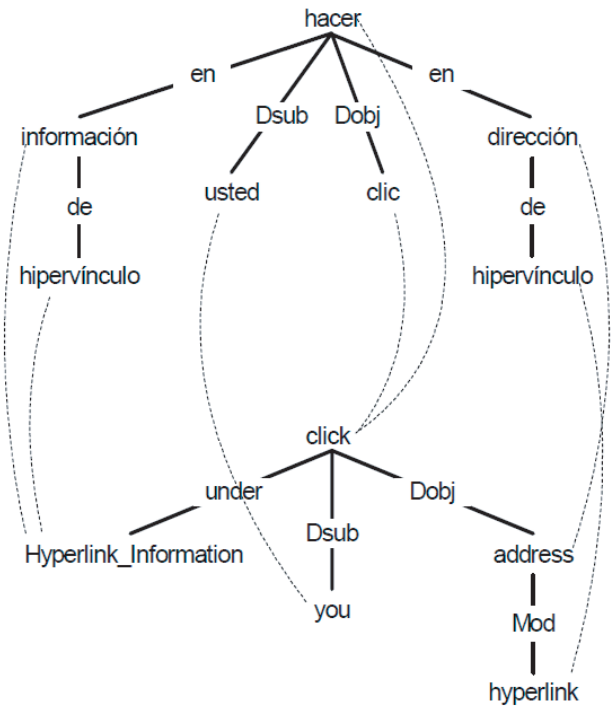

Figure 4: Alignment example of Spanish-English LF.

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 [12] の研究がある。

この研究は、隅田 [10] 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。

5.2.1 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。

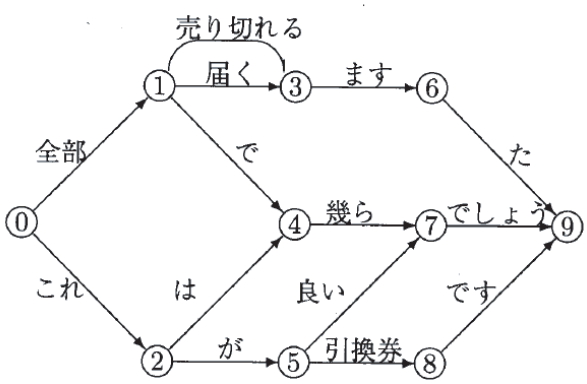

Figure 5: Example of word graph.

5.2.2 単語グラフ

各グループに属するすべての候補文は、図 5 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。

5.2.3 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。候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 構造的言語処理を指向するEBMTシステム

最後に、我々が研究してる用例ベース翻訳システムを紹介する。また、8 月に行なわれた IWSLT [1] という翻訳評価コンテストの結果も併せて紹介する。

[1] International Workshop on Spoken Language Translation (http://www.is.cs.cmu.edu/iwslt2005/)

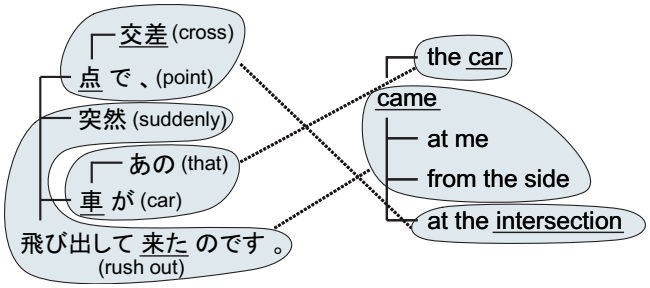

Figure 6: Example of alignment.

現在の機械翻訳の主流は統計翻訳であるが、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。

以上のような理由で、我々は用例ベース翻訳システムの研究を行なっている。我々の翻訳システムは、大きく分けてアラインメント部分と翻訳部分の 2 つに分けることができる。以下この 2 つのステップを順に説明する。

6.1 アラインメント

統計翻訳の基本姿勢は、対訳辞書や言語知識などは利用しないというものであるが、用例ベース翻訳ではこのような言語資源を積極的に利用する。

まず用例を得るための対訳文を、それぞれ図 6 のようなフレーズごとの依存構造を持った形に変形する。日本語文に対しては JUMAN[7] および KNP[6] を用い、英語文に対しては Charniak のパーサ [3] を用いて句構造に変換し、これにフレーズの head を定義するルールを適用することにより依存構造に変換する。

次に対訳辞書を用いて、単語ごとの対応をつける。対訳辞書としては、EIJIRO [2] [5] を利用した。図 6 では「交差点 - intersection」「車 - car」「来た - came」の 3 つの対応が得られている。単語ごとの対応がわかったら、その単語を含むフレーズをひとかたまりとして、フレーズごとの対応をつける。対応がつかなかったフレーズは、ルールによって他の対応のついているフレーズにマージする。図 6 の例では、「あの」が「車 the car」に、「突然」「at me」「from the side」が「飛び出して来たのです came」にマージされる。

このようにしてフレーズごとの対応付けを行ない、これらの対応すべてを用例としてデータベースに登録する。また複数の対応をまとめたものも、用例として登録する。図 6 の例からは、「交差点で、突然飛び出して来たのです」や「突然あの車が飛び出して来たのです」などを用例として登録する。これはできるだけ大きな用例を用いたいという用例ベース翻訳の考えによるものである。

6.2 翻訳

翻訳では、日本語入力文を依存構造に変換し、各部分に対して利用できる用例を検索し、その中からよい用例を選択して、英語文を組み合わせて翻訳を生成する (図 7)。

6.2.1 用例検索

まず依存構造の根を検索のルートノードとし、マッチするものがなくなるまで順に、ルートノードから大きな部分木を検索する。次に用例検索のルートノードを 1 レベルさげて、つまりルートノードの子供にあたるノードをルートノードとして同じことを行い、これを繰り返す。

図 7 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。

入力文のあるノードについて、まったく用例が検索できない場合には、対訳辞書を用いて訳を生成し、これを用例と同様にして扱う。

[2] http://www.eijiro.jp/

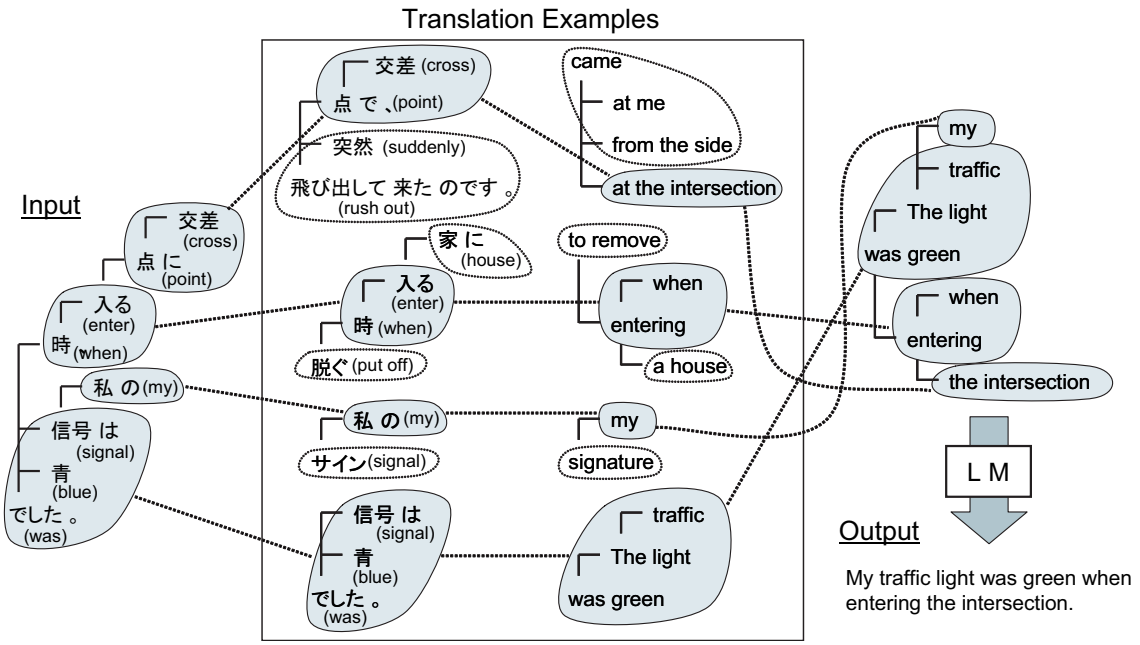

Figure 7: Example of translation.

6.2.2 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。このとき、用例ベース翻訳の基本的な考え方にもとづき、大きな用例を優先的に選択する。

用例の大きさの評価は、まずマッチするノード数を基本とし、その外側のノードについて類似する語である場合には類似度に応じて 0〜1 のスコアを与える。このスコアが大きいものを優先して選択する。外側のノードは用例を構造的に張り合わせるための「のりしろ」としても利用する。

たとえば、マッチ部分の大きさが 2 で、外側の親ノードが 0.3、一つの子ノードが 0.4 で類似している用例のスコアは 2.7 となる [3] 。

6.2.3 用例の組合わせ

翻訳文を生成するには、選択した複数の用例を組み合わせる必要がある。用例を組合わせるとき、多くの場合は上で述べたように、用例の外に糊しろがあり、その部分に用例を張り合わせることによって実現できる。

一方、まったく糊しろがない場合には、適当なルールによってコントロールしている。しかしこのような場合は多くはなく、影響は少ないと言える。

[3] これを翻訳確率で行う方法も提案しているが、実装は今後の課題である

Table 1: Evaluation results.

	BLEU	NIST
Development 1	0.4245	8.5655
Development 2	0.4056	8.4967
IWSLT05 manual	0.3718	7.8472
IWSLT05 ASR	0.3361	7.4157

また、日本語と英語の翻訳の場合には、日本語の省略が大きな問題となる。特に対話においては日本語の代名詞は省略される場合が多く、これが翻訳の誤りに結びつく。本来は日本語の省略解析などを用いて対応するべきであるが、まだ実装していない。現在は英語の言語モデルを用いて、この問題に対処している。言語モデルの学習には CMU の Cam_Toolkit[4] を用いた。

このようにして複数の用例を組み合わせることにより、翻訳を実現している。

6.3 IWSLT での結果

IWSLT では旅行ドメインでの音声翻訳を想定している。トレーニングセットとして 20000 文の旅行対話対訳コーパスが与えられ、ここから翻訳の学習 (EBMT においては用例の獲得) を行ない、506 文と 500 文の 2

つのディベロップメントセットによって翻訳システムを改善し、500 文のテストセットによって各システムの翻訳性能の比較を行なう。昨年まではすべてのデータは人手により書き下されたものであったが、今年は人手によるデータ (manual transcription) と、同じ文章の音声認識の結果を修正せずにテキスト化したもの (Automatic Speech Recognition:ASR) の 2 種類が用意された。性能は翻訳システムの自動評価尺度である BLEU や NIST などにより評価される。

結果は表 1 のようになった。他の参加システムと比較すると、精度的には真ん中より少し上位といった程度であった。

7 まとめと今後の課題

本輪講では、機械翻訳のおおまかな歴史について述べ、2 種類の主な機械翻訳手法である「統計翻訳 (SMT)」と「用例ベース翻訳 (EBMT)」の基本的な考え方を説明した。さらに用例ベース翻訳システムの関連研究について述べ、最後に我々が研究している用例ベース翻訳システムを紹介した。

今後は、活発に研究されている機械翻訳についての調査を続けるとともに、我々の研究している翻訳システムの精度向上を目指したい。具体的には、言い換え表現を吸収する柔軟な翻訳システムの実装と、構文解析誤りの動的な修正手法の研究を行なう予定である。

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*, pages 219–226, 2005.

[2] Peter F. Brown, Stephen A. Della Pietra, Vi cent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 1993.

[3] 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)*, pages 173–180, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

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

[5] Electronic Diciotionary Project. *EIJIRO 2nd Edition.* ALC Press Inc., 2005.

[6] Sadao Kurohashi and Makoto Nagao. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534, 1994.

[7] 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*, pages 22–28, 1994.

[8] 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*, pages 39–46, 2001.

[9] 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.

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

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

[12] 土居 誉生, 隅田 英一郎, and 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**, 45(6), 2004.