機械翻訳
概論(J-to-Eを中心に) & 統計的機械翻訳

言語情報科学論A

2007.07.30

林 良彦教授

Text: Courtesy of Dr. Jurafsky, D. and Dr. Martin, J.H: “Speech and Language Processing,1st edition (Prentice Hall, 2000) & 2nd edition”, http://www.cs.colorado.edu/~martin/slp2.html

機械翻訳(MT)の研究開発の歴史

- IBM，ジョージタウン大学のロシア語から英語への翻訳実験 (1954)
  - ハードウェアの進歩により機械翻訳は容易に実現できると考えられていた
- ALPACレポート (1965)
  - 機械翻訳の限界を指摘．研究開発が低迷化
- 多言語社会での成功例 (1970年代)
  - SYSTRAN (ヨーロッパ言語間)
  - TAUM METEO (モントリオール大学．天気予報の英仏翻訳)
- “Intelligence”における必要性 (2001年以降)

日本におけるMT研究の歴史

- 九州大学 (1955)，通産省電気試験所(現在の産業技術総合研究所．1956)
  - 日本語の問題 (漢字処理)
- 京都大学 長尾研究室 (1980年代~)
  - Muプロジェクト (電子技術総合研究所，日本科学技術情報センターなどとの国家的な共同プロジェクト)
  - 現在の多くの商用の翻訳システムの源流
- NTT (1984年~)
  - ALT-J/E：大規模な言語データ(辞書)を持つ日英翻訳システム
- インターネットの発展とともに(1995年~)
  - PC上の安価なシステムの出現

機械翻訳: 異言語間の対応をとる

- 語順の問題
  - 日本語:SOV, 英語:SVO
- 品詞の対応
  - 三冊の本 (数詞＋助詞＋名詞)
  - three books (形容詞＋名詞)
- 意味の多義と訳し分け (例:かける, play)
- 文化背景の違いによる語概念の違い
  - beef <-> 牛肉
- しかし, 基本は**要素合成原理 (compositionality)**

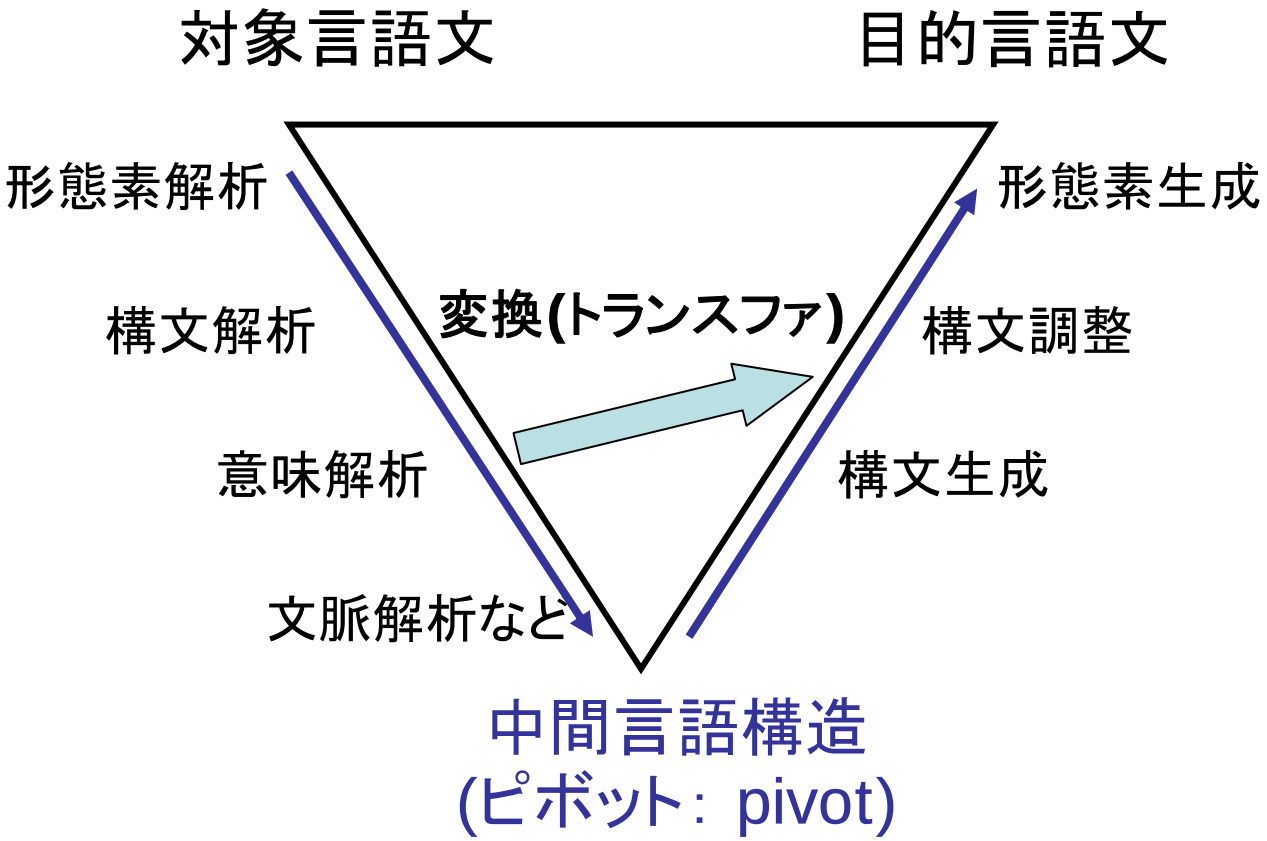
要素合成原理

- 文の文法的構造
  - 文 = 文 + 接続属性 + 文 | 単位文
  - 単位文 = モダリティ情報 + 命題情報
  - 命題情報 = 格要素* + 述語句
  - 格要素 = 名詞句 + 格助詞表現
  - などなど
- 基本原理: 要素合成原理
  - それぞれの部分要素を翻訳した結果を組み立てる(合成)することにより，全体の翻訳を得る

機械翻訳の基本方式

- 単語直接置き換え方式
  - 形態素レベルの解析・生成＋対訳辞書
- トランスファ方式
  - 対象言語(SL: Source Language)文を解析し，SL言語に依存した構造を作る
  - SL言語構造をそれに対応した目的言語(TL: Target Language) の構造へと変換する (= トランスファ)
  - TL言語構造からTL言語文を生成する.
- 中間言語方式
  - SL言語を解析し，特定の言語に依存しない構造(中間言語構造)を作る
  - 中間言語構造からTL言語文を生成する.

トランスファ方式と中間言語方式



トランスファ方式と中間言語方式の比較

- トランスファ方式
  - 最適な変換レベルを言語対に応じて設定することができる（例：英仏翻訳 vs 英日翻訳)
  - 多言語化のためには，言語対に応じた変換処理を作る必要がある
- 中間言語方式
  - 多言語化に有利．(変換処理が不要である)
  - 中間表現の設計は困難
  - 中間表現の設定によっては，ニュアンスなどの情報が失われる可能性がある

変換規則の観点からみた機械翻訳の基本方式

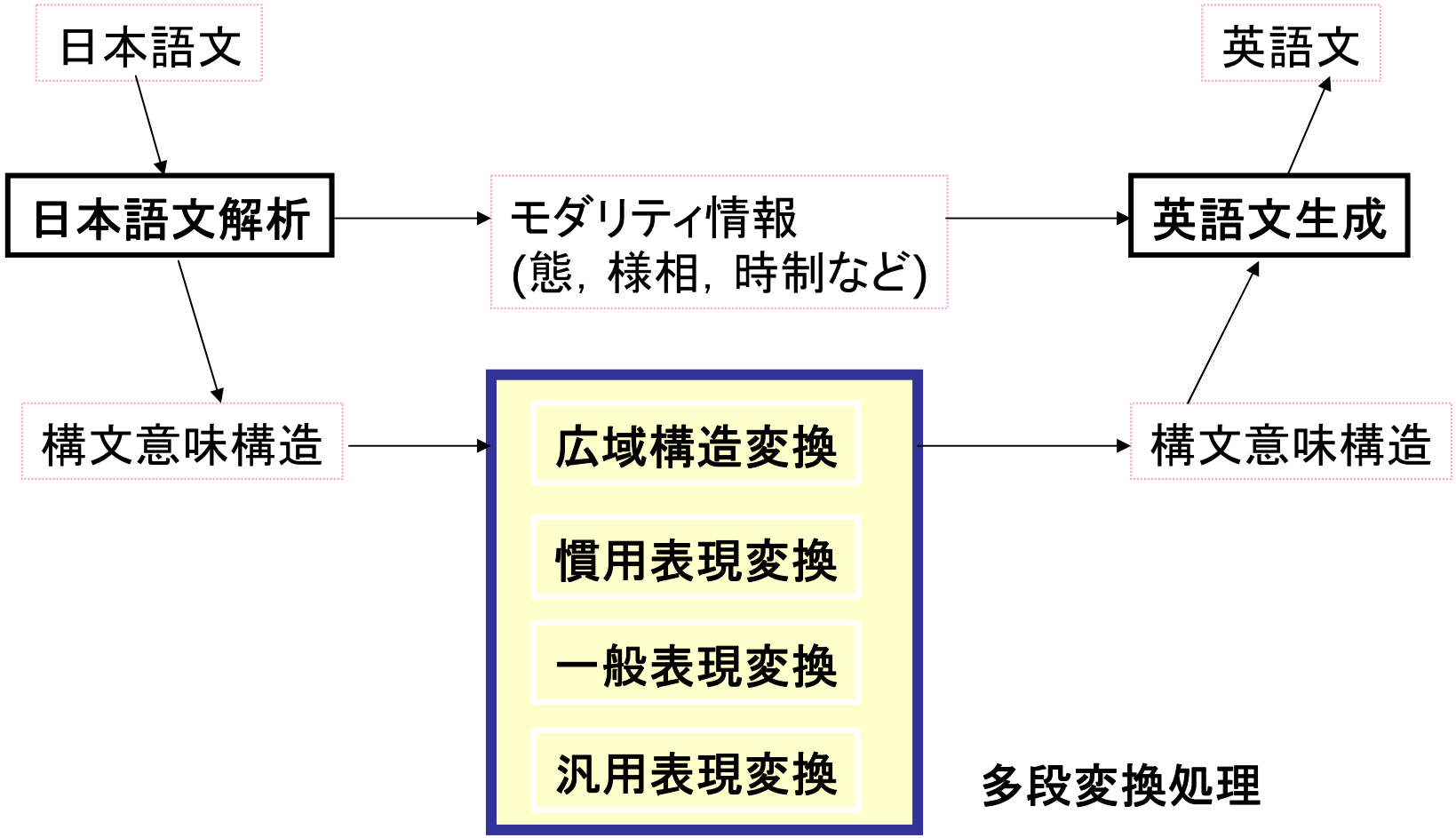
- ルールベース方式
  - 人手で変換規則を構築する
- 用例ベース方式
  - あらかじめ翻訳例を蓄積しておく．入力と似ている翻訳例を検索し，それを利用することにより翻訳する
- パターン/テンプレートベース方式
  - 定型的な表現に対して，翻訳パターンをテンプレート化しておき，変数部分だけを入れ替えて翻訳する
- ハイブリッド方式
  - これらの組み合わせ
- 統計ベース

トランスファ方式による日英翻訳

- ALT-J/Eをケーススタディとして -

- 形態素解析 → 構文解析 (係り受け解析) → 意味解析
- 変換処理
  - 構文意味構造変換
  - 名詞句変換
  - 副詞の変換
- 英文生成
  - 副詞句の語順
  - 数の生成
  - 決定詞の生成

構文意味構造変換


日本語文
英語文
日本語文解析
モダリティ情報
(態，様相，時制など)
英語文生成
構文意味構造
広域構造変換
慣用表現変換
一般表現変換
汎用表現変換
構文意味構造
多段変換処理

広域構造変換

- 例：彼は走って家に帰った
- 普通に翻訳すると
  - He ran and returned to home.
- 「走って [場所]に 帰る」←→ run back to ~
  という単文の枠を越えた変換を行う
  - He ran back to home.
- 複文をパターン化することで，ある分野における定型的な表現を高精度に翻訳することができる.
- 要素合成原理を越える試みの一つ

慣用表現変換

- 例：私は彼のしっぽをつかみ，彼は猫のしっぽをつかんだ.
- N1[主体]が N2[主体]の しっぽを つかむ ←→ N1 find N2’s weak point
- N1[主体]が N2[具体物]を つかむ ←→ N1 grasp N2
- ポイント: 慣用的な表現を優先してチェックする
- 訳出：I found his weak point, and he grasped the cat’s tail.
- ALT-/JE では 約3,000の慣用表現パターンをルール化

一般表現変換
結合価パターン変換

- 例: 店員はお客から料金を取った
- N1[主体]が N2[料金]を N3[主体] から/より 取る ←→ N1 charge N3 N2

- 例: 社長が総会で挨拶する
- N1[主体]が N2[行事]で 挨拶する ←→ N1 give an address at N2
- *受身形にはできない!*

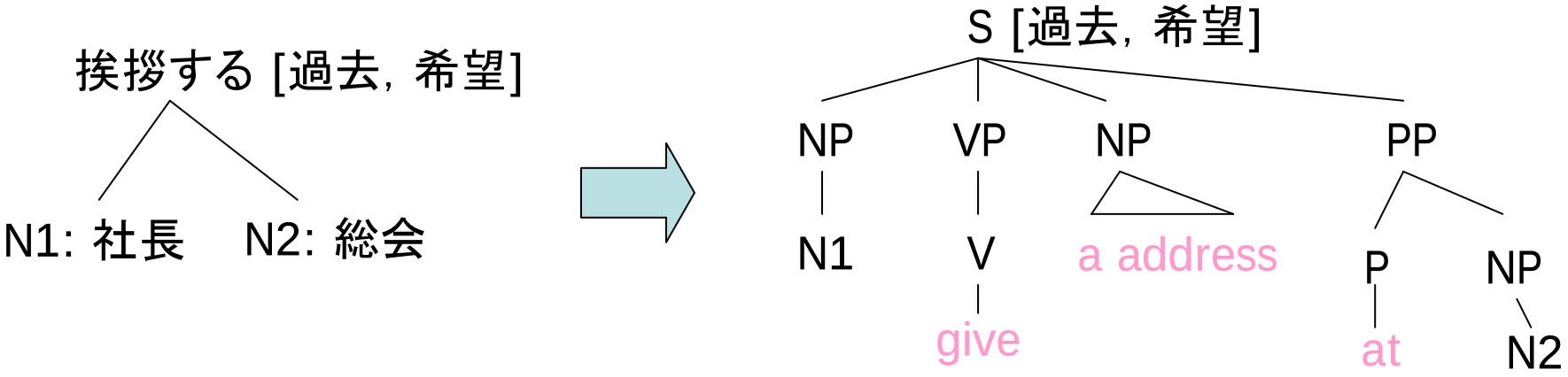
- ALT-/JEでは，約6,000語の日本語用言に対して，約13,000のパターンをルール化

汎用表現変換

- 慣用表現変換，結合価パターン変換できなかったもの(辞書・ルールの不備) を救済
  - N1が N2を V(他動詞) ←→ N1 V N2
- Copulaの変換
  - N1が N2だ ←→ N1 be N2
  - それが問題だ ←→ It is the problem.

構文意味構造変換のまとめ

- 広域表現変換: 単位文(述語句)が固定
- 慣用表現変換: 格要素のいくつかが固定
- 一般表現変換: 述語句のみが固定
- 汎用表現変換: 述語句も変数
- 結果の表現例:「社長が総会で挨拶したかった」



名詞句の変換

- 基本的には要素合成原理を適用
- それではうまくいかない例：象の鼻 ←→ trunk (×: elephant’s node, nose of elephant)
  - 辞書の属性で対処

- 複合名詞 (compound nouns)
  - 基本は辞書登録．構造的な特徴で分類
    - 数量表現型
    - 固有名詞型
    - 機能語型：接辞(~用，~製，~型) の性質を利用してルール化

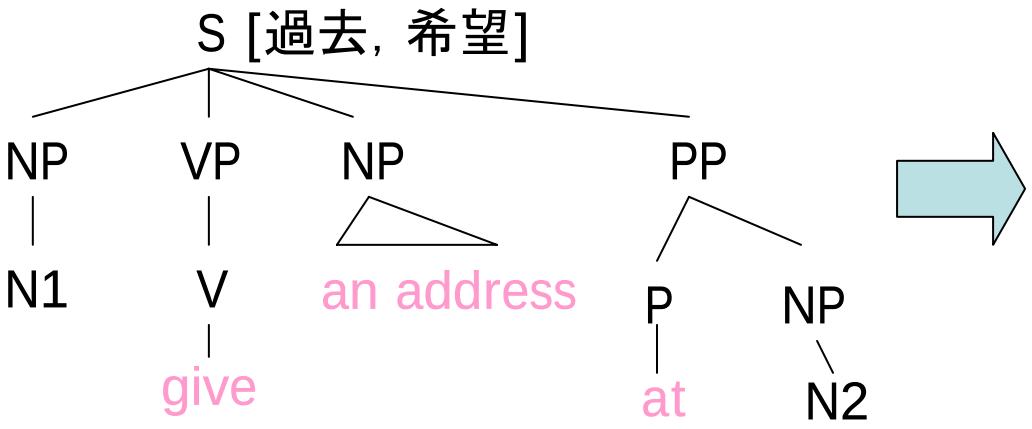
副詞の変換

- 例: よく
- よく知っている ←→ know ~ well
- よく聞く ←→ listen ~ carefully
- よく訪ねる ←→ visit ~ often

- 決定規則：用言の意味属性，様相，時制などを組み合わせて決定する
- 自然な訳出を行うために，英語コーパスでの共起情報も利用

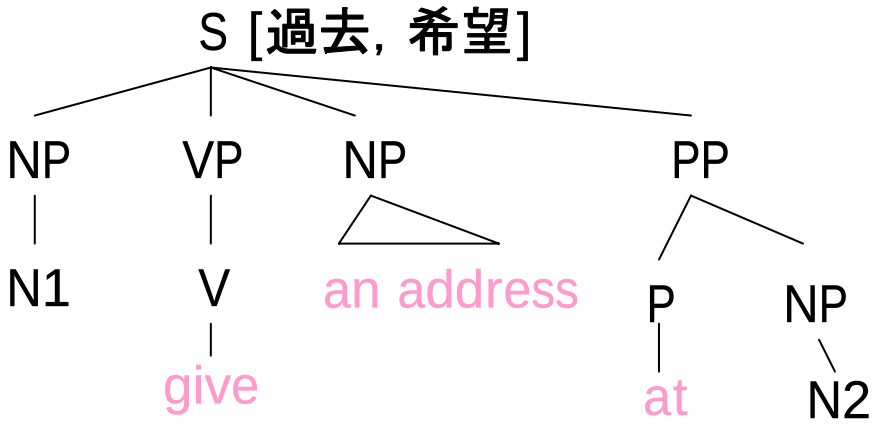
英文生成

- 図のような構文意味構造から，英語文の文字列を導く
- 様相・時制情報の処理
- 副詞の語順決定
- 数量表現の決定
- 決定詞の生成



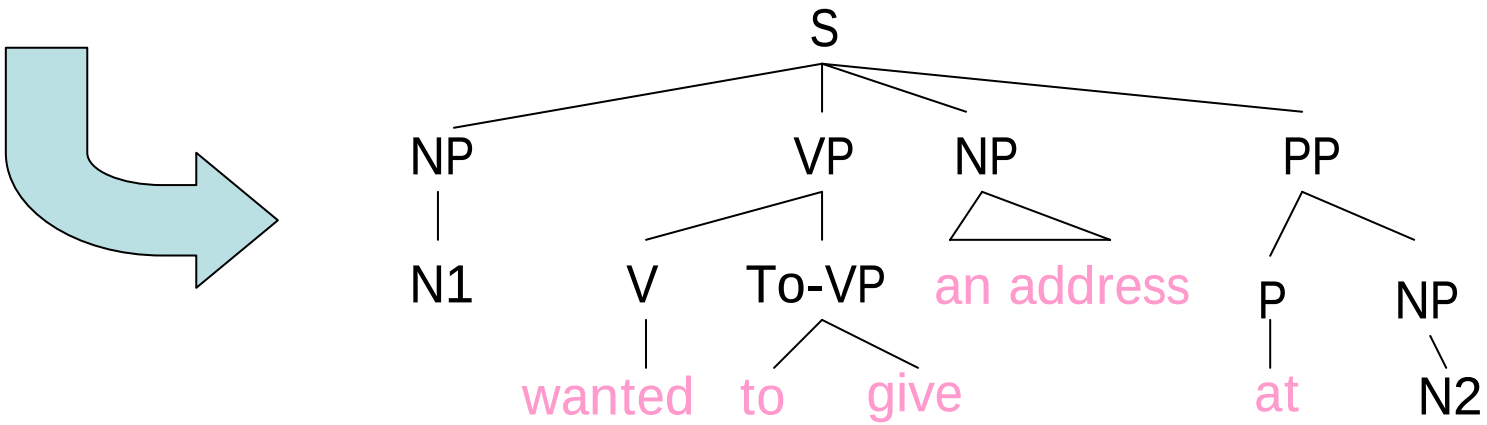
The president wanted to give an address at the general meeting.

時制・様相情報の処理



まず「希望」の様相を処理し，
次に「過去」の時制を処理する

英語→英語の構造変換



副詞の語順決定

- 英語の副詞を細かく分類しルール化
  - 文法的 (付接詞:様態，場所，時間，従接詞：強調，離接詞：話してのムード，合接詞：接続詞的なもの)
  - 意味(様態，場所，時間，頻度，強意)
  - 優先的な語順位置(文頭，中位，文末，後置，前置)
  - “Morally, it is not right.”
  - “This failure is clearly his responsibility.

数量表現の決定

- 加算名詞/不加算名詞の区別を5段階に分類し，
- 名詞の属性との関係でルール化

- “I bought two books.” (通常の加算名詞)
- “She ate two slices of cake.” (粉砕的)
- “They gathered flowers.” (集合的)
- “Mammonths die out.” (総称的)
- “All children become adults.” (帰属的)

決定詞の生成

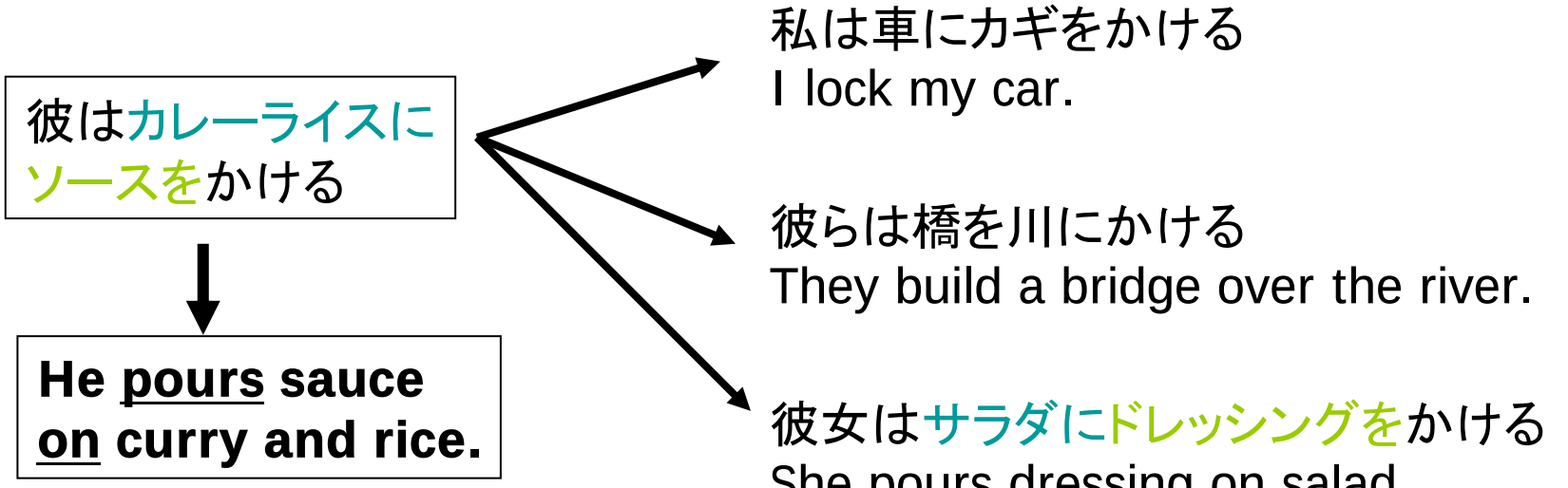
- 総称的：無冠詞
- 帰属的：複数のときは無冠詞，単数のとき不定冠詞(a)
- 指示的：「定」のとき，他の決定詞(this, some など)がなければ定冠詞(the).
  「不定」のとき，複数なら無冠詞，単数なら不定冠詞
- 「定」と「不定」の区別： 文脈解析が必要

「定」であることが文脈解析なしに決まる例

- 連体修飾節: “the house whose roof is red”
- 特定の修飾句: “the fastest train”
- 場所を示す名詞句
- 確定文(AはBだ)の主語: “The registration fee is 30,000 yen.”

用例ベース方式 / Example-Based MT

- 用例データベースを作成：対象言語と目的言語を文，句，単語の各レベルで対応付けておく必要がある
- 入力文に類似し用例をデータベースから検索する
- *カバー率を向上させるためには，用例をある程度抽象化して整理しておく必要がある*



パターン/テンプレートベース方式

- ALT-FLASH (市況速報をターゲットとしたハイブリッド型翻訳システム)
- テンプレート翻訳の例
  日本語: /売買高/商 /は/が/ /概算/概算で/約/ <<*数値1*>>株[だった] ←→
  英語: Trading volume was estimated at <<*1*>> shares.
- どうやってテンプレートを収集するか
  - コーパスからよく現れる n-gram を高速に抽出する処理によりルール化を支援

機械翻訳の評価と利用

- 訳文の品質評価
  - 正確性，理解容易性，忠実性
  - 重要な点は，原言語文なしに訳文を修正することができるかどうか.
  - ベンチマークテストの設定の必要性 (ALTでは6,200文の機能試験用の文を作成)
- 機械翻訳の利用
  - 前処理 (pre-edit)：翻訳しやすいように原言語文をあらかじめ修正する
    - 長文の分割，係り受け関係の明示，省略を補完するなど
  - 後処理 (post-edit)：翻訳された文を編集する

機械翻訳の課題

- 頑健性 (robustness) の向上
  - 対象言語の解析における失敗を防ぐ
  - 辞書，文法のカバー率を向上させる
  - 長文，並列名詞句の解析率の向上
- 訳文品質の向上
  - 用例ベース，パターン/テンプレートベースの翻訳を通常のルールベース翻訳とうまく融合させる
- 言語知識 (辞書，ルール) の整備，メンテナンス
  - コーパスに対する統計的処理による支援
- 実用における支援環境
  - 前編集/後編集，辞書登録，別解釈表示など

The Tower of Babel

昔は世界の言語はひとつだったが，天まで届こうかという塔を築こうとする
人間の尊大さをこらしめるため，神が人間の言葉を乱した．これにより，
意思疎通が難しくなり，塔の建設も中止せざるをえなくなった．
人々は散り散りになり，それにしがたい使う言葉も分かれていった．(大意)

– 参照：旧約聖書創世記
  第11章

財団法人　日本聖書協会
http://www.bible.or.jp/main.html

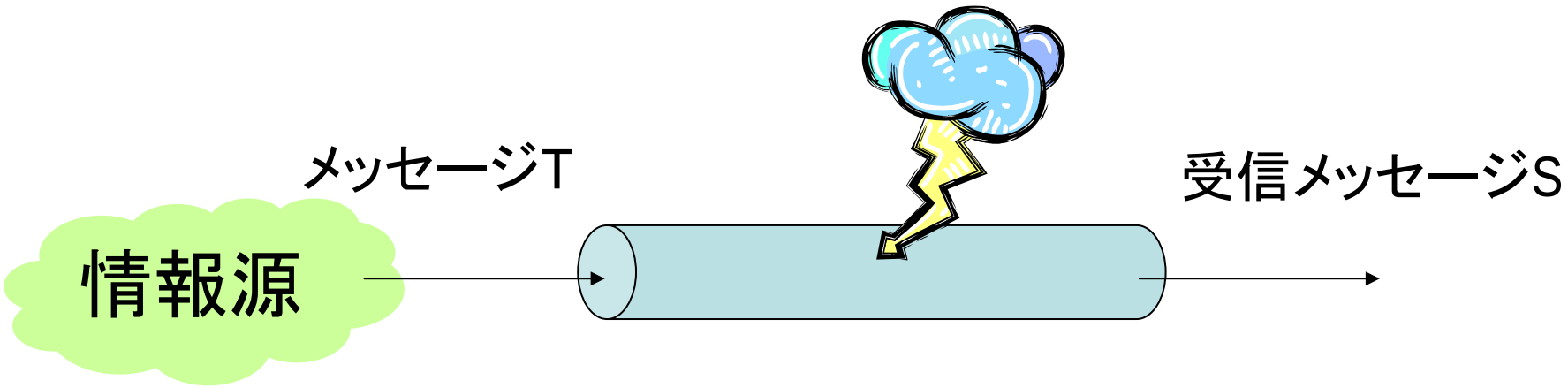
– Pieter Bruegel
  1563

変換規則の観点から見た分類

- ルールベース方式
- 統計的機械翻訳 (SMT; Statistical Machine Translation)
  - 大量の対訳コーパスから，2言語間の翻訳の仕方を自動的に学習する
  - 翻訳の仕方
    - 入力中のある単語・句は，相手言語でどのような単語・句に対応するか
    - それらは，相手言語において，どのような順に並ぶか

2003年6月11日付 Wired News記事 ― WIRED VISION (http://wiredvision.jp/)より

Noisy Channel Model



情報源のモデル
P(T)

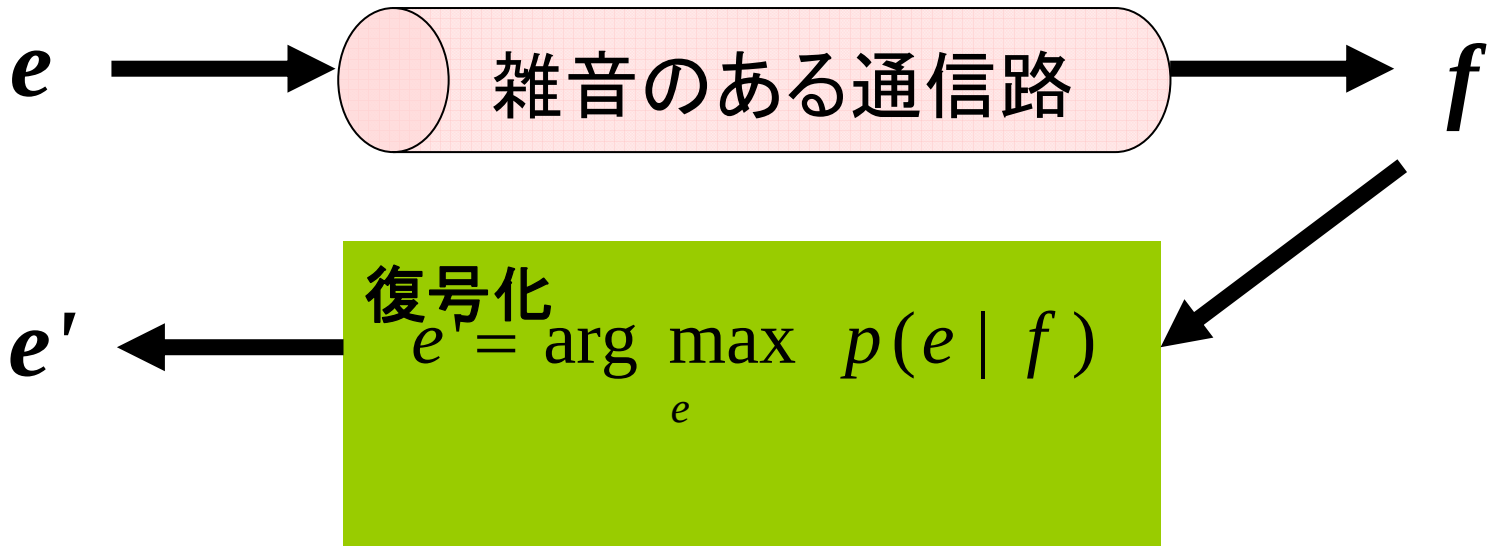
伝送路のモデル
P(S|T)

- これまでに発生したメッセージTの事例からP(T)を推定する
- これまでに受信したメッセージSと
  対応するオリジナルのメッセージTのペアの事例から
  P(S|T)を推定する

復号化問題: あるメッセージSを受信したとき，そのオリジナルのメッセージTを求める

SMTの基本的考え方: Noisy Channel Model

- Noisy Channel Model: 情報理論 (after Shannon)
- 仏英翻訳の例: 英語を流すとフランス語になってしまう不思議な(noiseだらけの)通信路があるとする
- 仏英翻訳とは: 観測されるフランス語文f から，そのもとになっている英語文e を推定する(復号化する)問題



p(e|f) をどうやって計算するか

- 前回とおなじくBayesの定理により変形

$$e' = \arg\max_{e} p(e \mid f) = \frac{p(e) \bullet p(f \mid e)}{p(f)}$$

- 分母 P(f) は最大確率を与えるe を求めるのには関係ない．よって，

$$e' = \arg\max_{e} p(e \mid f) = p(e) \bullet p(f \mid e)$$

言語モデル確率: どのくらい英語らしい文か

翻訳モデル確率: 言語間の対応はどのくらい良いか

Courtesy of Dr. Jurafsky, D. and Dr. Martin, J.H: “Speech and Language Processing, 2nd edition”, CH. 24
http://www.cs.colorado.edu/~martin/slp2.html

SMTに必要なことは結局...

- 言語モデル確率 p(e) の計算
  - 英語のコーパスがあれば計算できる (n-gram model)

$$p(w_1, w_2, \cdots, w_n) = \prod_{i=1}^{n} p(w_i \mid w_{i-N+1}^{i-1})$$

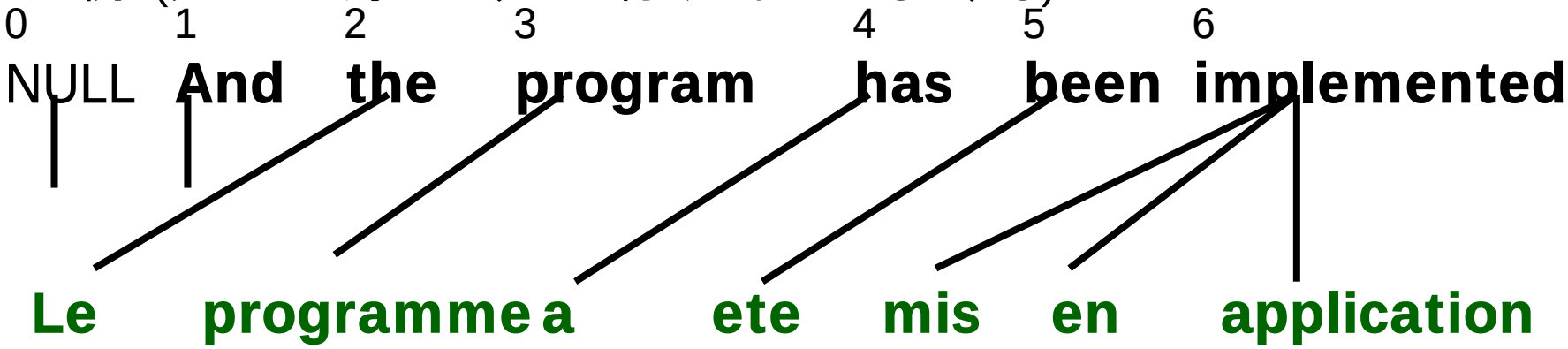
- 翻訳モデル確率 p(f|e) の計算
  - 対応付け (alignment) a を考えて次式で計算

$$p(f \mid e) = \sum_{a} p(f, a \mid e)$$

  - 英文e の翻訳がフランス語文f とする．このとき，単語対応には多くの可能性があるが，翻訳モデル確率は，これらの全ての単語対応の可能性について，条件付き確率P(f,a|e)の和をとったものと考える
- 復号化アルゴリズム: argmax p(e)p(e|f) の計算
  - 探索空間は非常に膨大
  - 近似的な探索アルゴリズム (stack decoding)

対応付け (word-for-word alignment)

- 例: (文ごとの対応はすでに付けられているとする)



——— 接続 (connection) と呼ぶ　NULL: 仮想的な単語

  - 上記の対応付けのベクトル表示: [2,3,4,5,6,6,6]
- 対応付けの制約 (IBM Model)
  - フランス語の各単語はどれか一つの英語の単語と対応
    - 複数の英語単語がフランス語の一単語に対応するのは認めない
    - 接続を持たないフランス語の単語は，NULLに接続と仮定
  - 英語文の長さをl , フランス語文の長さをm としたとき，可能な対応付けの個数は $(l+1)^{**}m$ に制限できる

p(f,a|e) をどうやって計算するか

- 翻訳モデル (IBM Model-1)
  - フランス語の単語fj は，接続された英語の単語eaj のみに依存して決まる
  - フランス語の文の長さの確率はある定数 ε とする
  - 全ての単語対応は等確率で英文の長さl のみに依存する

$$p(f, a \mid e) \propto \prod_{j=1}^{m} t(f_j \mid e_{aj})$$

翻訳確率: ある対応付けa において
英語単語ej がフランス語単語fj に接続する確率

t(f|e) をどうやって計算するか

- "Chicken and Egg Problem"
  - コーパス中の単語レベルの対応付けが分かっていれば, t(f|e) は計算できる
  - 逆に, t(f|e) が分かっているなら, コーパス中の単語レベル対応付けを見出すことができる
- 対訳コーパスを用いてEM (Expectation Maximization) 法と呼ぶ反復計算により推定する
  - EM法: この手の問題における定石手法
  - アルゴリズムの概要
    - 初期値割り当て: ある英語の単語は, 全てのフランス語の単語に等しい確率で対応しうると考える. 例えばフランス語の語彙サイズが10,000語なら, t(maison|house) もt(velo|house) も1/10,000である
    - 対応付けの確率計算: 上記の値をもとに, 全ての可能な対応付けの確率を求める
    - 期待生起回数の計算 (Expectation Step): この対応付けの確率をもとに, 各接続の期待生起回数 (fractional countと呼ぶ)を計算する
    - 翻訳確率の再計算: 期待生起回数をもとに翻訳確率を再計算する
    - 収束条件を満たすまで繰り返す

EM法の例

- 簡単な例 (NULLは考えない. 英仏の1:n対応も考えない)

	s1	s2
english	b c	b
french	x y	y

対応付け: a1(b-x, c-y), a2(b-y, c-x), a3(b-y)

- Step-1 初期化 (一様な初期値を与える)
  - t(x|b) = 0.5, t(y|b) = 0.5, t(x|c) = 0.5, t(y|c) = 0.5
- Step-2 p(f, a|e)を全ての対応付けについて計算
  - a1(b-x, c-y): p(f, a1|e) = 0.5×0.5 = 0.25
  - a2(b-y, c-x): p(f, a2|e) = 0.5×0.5 = 0.25
  - a3(b-y): p(f, a3|e) = 0.5
- Step-3 それぞれを文単位でnormalize
  - s1 について p(f, a1|e) = p(f, a2|e) = 0.5
  - s2 について p(f, a3|e) = 1

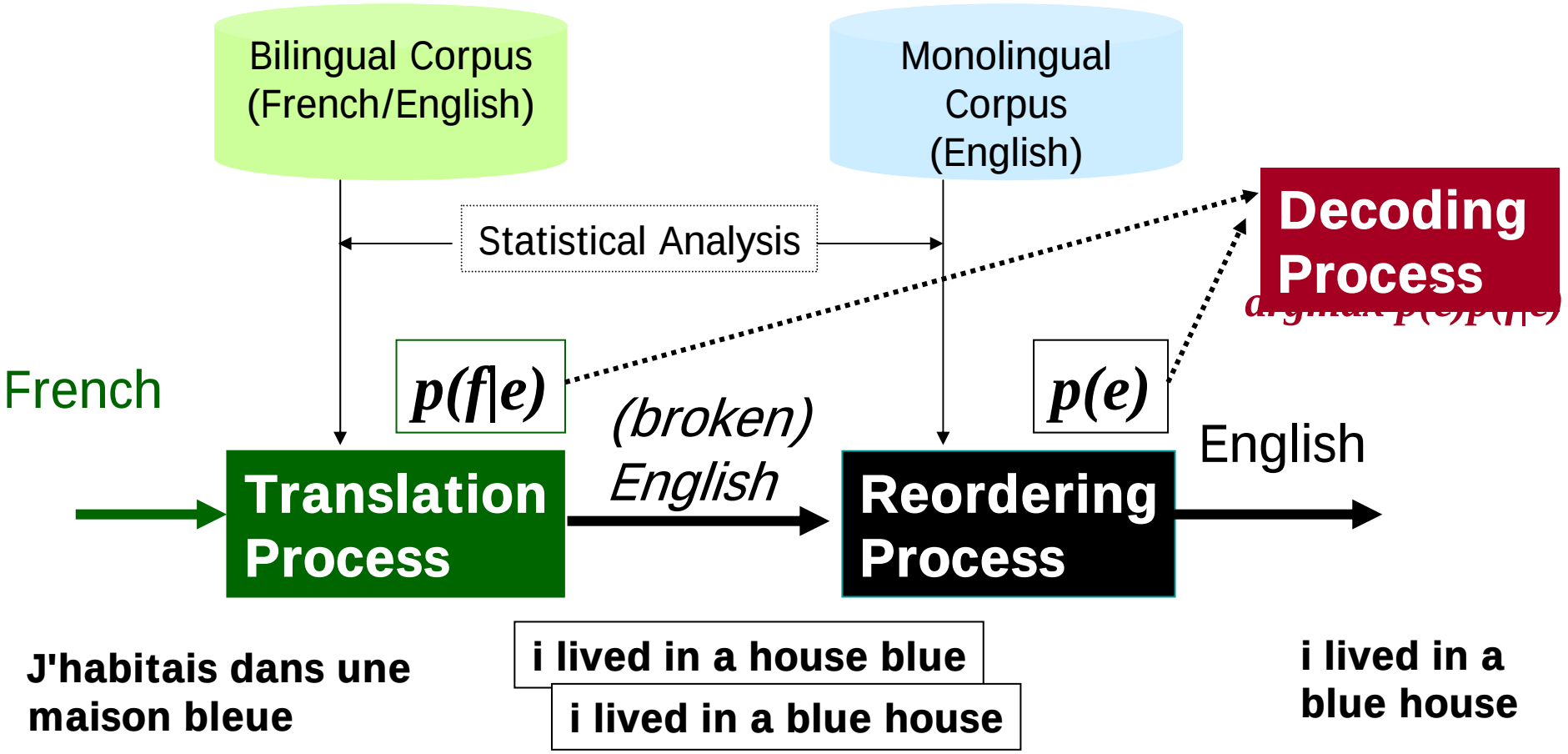
EMの例 ~cont.

- Step4 期待接続回数を修正（Expectation Step)
  - b-x, c-x, c-y の接続: 0.5回
  - b-y の接続: 0.5 + 1 = 1.5回
- Step5 これをもとにt(f|e)を再計算
  - t(x|b) = 0.5 / (0.5+1.5) = 0.25
  - t(y|b) = 1.5 / (0.5+1.5) = 0.75
  - t(x|c) = 0.5 / (0.5+0.5) = 0.5
  - t(y|c) = 0.5 / (0.5+0.5) = 0.5

0.125
0.875
0.75
0.25

- Step2 ~ Step5 を繰り返す
  - 収束条件: 繰り返してもある定数以上の変化がない
- 最終結果例
  - t(x|b) = t(y|c) = 0.0001
  - t(y|b) = t(x|c) = 0.9999

Another View of SMT



Remark: Translation Processは出力される英語が文法的であるとか，こなれた文であるかなどは全く考慮していない．それは，Reordering Processの仕事である!

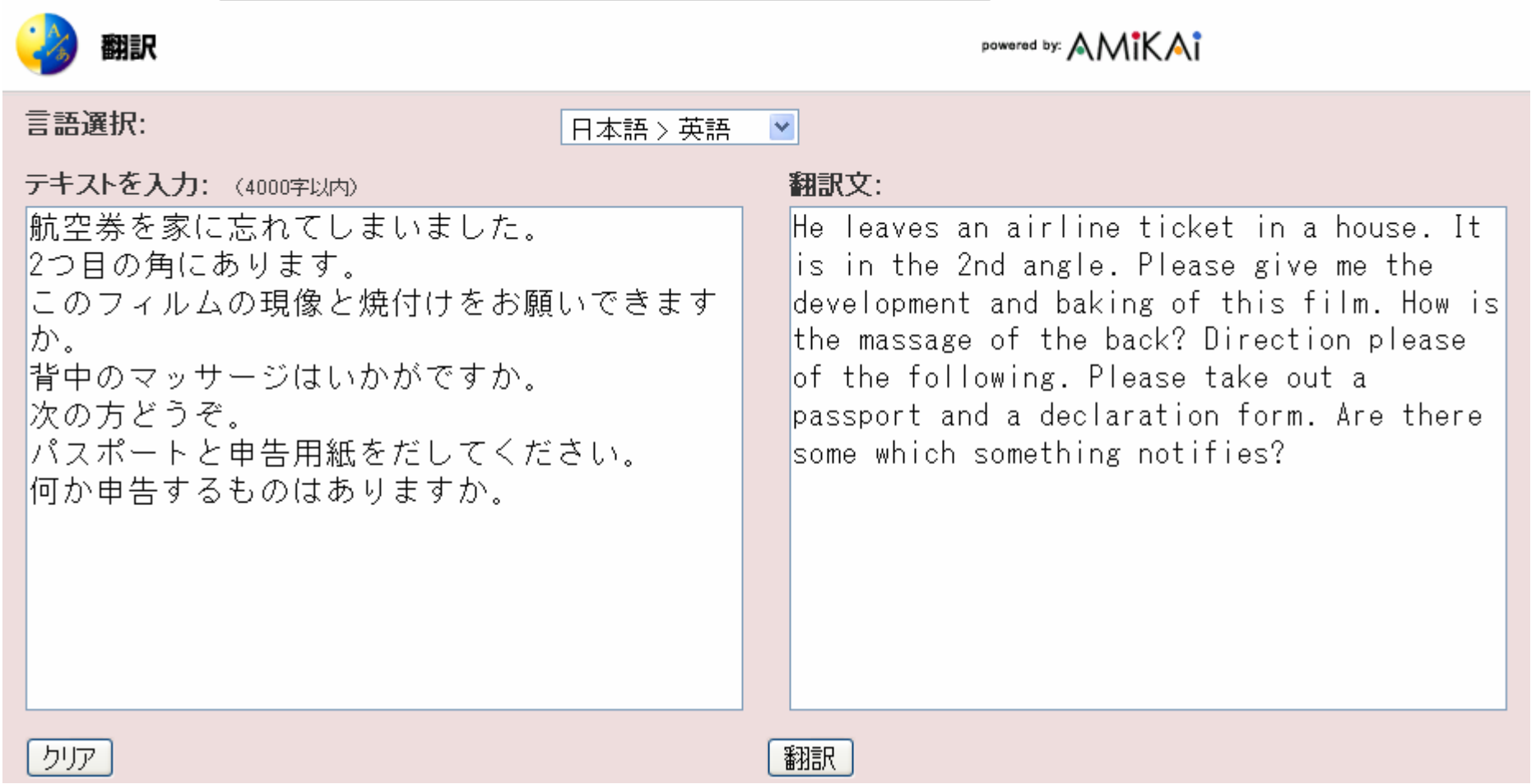
SMTの実際

- 翻訳モデルの詳細化 (IBM Model-3 ~)
  - 翻訳確率
  - 繁殖(fertility) 確率 (1:nの単語対応を扱うため)
    - p(n|e) : 英語の単語eがn語のフランス語に対応する確率
  - 歪み(distortion) 確率 (語順の違いを扱うため)
    - d(j | i, l, m) : 長さl の英語の文における単語位置i の単語が長さ m のフランス語文において単語位置j の単語に接続される確率
  - ...さらに複雑なModel-4, 5もある
- 単語単位の対応付けの限界
  - 語順が大きく異なる言語対では，対応付けの可能性が爆発する
  - 単語列(chunk)や句(phrase)を単位とする対応付けの追求
- 復号化 (decoding) プロセス
  - 入力文の単語数に対して指数的に計算量が増大（NP完全問題)
  - 効率的な近似アルゴリズム: スタックデコーディング

SMTによる日英翻訳の例 (IWSLT2004から)

- 航空券を家に忘れてしまいました
  i left my ticket at home
- 2つ目の角にあります
  it's on the second corner
- このフィルムの現像と焼付けをお願いできますか
  can i develop and print this film
- 背中のマッサージはいかがですか
  would you like a facial massage on my back
- 次の方どうぞ.パスポートと申告用紙をだしてください. 何か申告するものはありますか
  please give me your passport and next person form do you have anything to declare

従来方式による翻訳 (どっちが勝ち?)



アミカイ株式会社ご提供

統計的機械翻訳の pros & cons

- 翻訳対象の文のジャンルや分野に対応したコーパスを用いることで，それに特化した高精度な翻訳が可能 (かもしれない)
  - 学習したコーパスに適合しない文はうまく訳せない
- 対訳データさえあれば*，言語学者は要らない
  - *実際には，品詞付けや分かち書きの処理が必要
  - 対訳データを集めるのは実は大変！
- 統計処理により翻訳の仕方を学習するので，規則の微調整の苦労がない
  - 手作業で修正を施してチューンアップすることができにくい
  - 言語学的な知見・知識を融合させにくい
- 英語-フランス語などでは高い精度を得ている
  - 日本語-英語などの語順や構造が大きく異なる言語では精度がでにくい．組み合わせが莫大で計算も大変！

機械翻訳システムの評価

- 機械翻訳システムの評価
  - 翻訳速度
  - 分野への適用性
  - 翻訳品質
- 翻訳品質の客観的評価の必要性
  - 利用者: より良いシステムを求めるため
  - 開発者: より良いシステムを作るため
- 翻訳品質の評価
  - 従来は，人手(専門家)による評価が必要であると考えられていた
  - 自動評価指標の提案: 統計的翻訳において，パラメータを最適化するための指標として，自動評価点を利用

人手による評価の観点

- 流暢さ (fluency)
  母国語とする人にとってどの程度自然か
  - 5:問題なし, 4:良い, 3:非母国語的, 2:不自然, 1:理解不能
- 適切性 (adequacy)
  原文の情報がどの程度含まれるか
  - 5:全ての情報, 4:ほとんどの情報, 3:多くの情報, 2:少しの情報, 1:情報なし

- 問題点
  - 同一評価者: 評価のゆれ
  - 評価者間: 評価のバラツキ

自動評価

- 用意するもの
  - 原文: テストデータ
  - 参照訳 (references): テストデータの各分について，典型的な翻訳文を複数(4~16通り)する
- 自動評価指標 BLEU (2001) の基本的な考え方
  品質が良い訳文と参照訳とは文中の単語列が頻繁に一致するが，品質の悪い訳ではそうならない
- BLEUの計算式: 次ページ
- BLEUの特徴
  - 人手による評価と高い相関 (0.8前後~0.95程度)
  - 特に，流暢さに関して高い相関 (通常は4-gramを用いるため，フレーズに近い単位で評価を行うことに相当するため)

BLEUの計算式

$$BLEU = BP \bullet \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

- 以下のBP とpn の加重(wn)幾何平均
  - BP : 短い翻訳文が高い評価点にならないように補正するパラメータ
  - pn : n-gram適合率．翻訳結果中のn-gramの総数(分母)のうち，参照訳中のいずれかのものに一致するもの(分母)割合
  - wn : n-gramをどの程度重視するかの重み
- 通常は，N=4．すなわち，4単語の連鎖までを考慮する

まとめ

- コンピュータのパワーの増大，言語データ(コーパス)の充実により，統計的なアプローチが盛んに試みられている．その利点は:
  - ルール作成のコスト削減・質の均一化
  - 理論的な裏づけ (heuristicsに基づくcost ⇒ 確率)
- 統計的機械翻訳の活発化・成功
  - より多くの対訳データが利用可能となった
    - SMTの精度は，対訳データ量に応じてよくなる (more data is better)
  - コンピュータハードウェアの進歩
    - 大量のデータを高速に処理可能
  - 自動的な評価指標が利用可能となった
    - 開発⇒テスト・評価⇒ のサイクル
  - 翻訳品質も人手によるルールベース方式を上回りつつある
    - ただし，日本語と英語のように語順が大きく異なる言語の間の翻訳にはさらに工夫が必要
      - まとまり(chunk, phrase) レベルでの処理

Noisy Channel Model, Revisited

$$T' = \arg\max_{T} P(T)P(S \mid T)$$

言語モデル　音声認識:音響モデル
機械翻訳:翻訳モデル

適用	*T (もとめたいもの)*	*S (観測できるもの)*
音声認識	単語列	音声波形
スペル訂正	正しい単語列	スペルミスを含む単語列
情報検索	検索要求 (query)	文書 (document)
文書要約	要約 (summary)	文書 (document)

参考:最適経路探索問題

- 最適経路問題
  - 可能な経路がグラフネットワーク構造で指定される
    - ノード，リンクに重み(コスト，得点，確率)が付与
  - 始点sから終点gに至る最適な経路を求める問題
  - 「しらみつぶし」に探索すれば答は求まるが，効率的な方法が存在する
- 最適性の原理
  - 最適経路中の部分経路もまた最適経路になっている
  - 任意のノードvにおいて
    - W(s, g) = W(s, v) + W(v,g)
- 動的計画法 (Dynamic Programming)
  - 最適性の原理に基づく効率的なアルゴリズム
  - 任意のノードvにおいて，複数の前段のノードからの経路の可能性があるが，最適なもの(どのノードからか，その時の重みはいくらか) のみを記録しておけばよい