コーパスベースの機械翻訳

内山将夫@NICT
mutiyama@nict.go.jp

- コーパスとは単言語もしくは多言語のテキストの集積のこと
- コーパスベースの機械翻訳とは，コーパスから
- 各種の知識を自動獲得し，それを利用して，
- 機械翻訳をすること

背景

- コーパスベースの機械翻訳 (MT) は
- 学部実験で扱えるくらいに
- 各種のツールが整備されてきた

この講義の目的１

- 個人で機械翻訳を研究したい人が
- ツールのアルゴリズムを理解し，それを
- 改善し，新たなアルゴリズムを作成できるだけの
- 基礎知識を提供すること

この講義の目的２

- 自分でオープンソースの MT システムを利用して,
- 機械翻訳の実験をするくらい興味を喚起すること

成績評価方法

次の3通りのいずれかにより，成績を評価するので，各自が，どの評価法による評価を望むかを指定し，当該のレポートを提出して下さい．

評価法1 (MT実験) オープンソースのMTシステムを，各自の計算機で実際に動かしてみて，その過程をレポートする．

評価法2 (課題回答) 講義スライド中にある課題のうちで，興味のある課題について，レポートする．

評価法3 (授業態度) 講義の前に，講義スライドを全て読み，講義スライドにおける疑問点等を事前にリストアップし，講義においては，疑問点を質問し，そのやりとりの過程を記録する．そして，これらの疑問点とやりとりについてレポートする．

講義の内容

1. オープンソースの MT システムの例

---

2. MT の性能評価についての一般的な話題
3. MT の性能をどう測定するか？ 実験の作法と MT の自動評価
4. 自動評価尺度 BLEU

---

5. 初歩の確率
6. 初歩の言語モデル

---

7. 単語対応の導入
8. IBM Model-1 の式の説明
9. IBM Model-1 の動作例

---

10. 現状で利用可能なパラレルコーパス
11. パラレルコーパスを利用した検索と対数尤度比検定による対訳抽出
12. パラレルコーパスの自動作成

---

13. 翻訳モデルと翻訳エンジン
14. 句単位の翻訳モデルの概要
15. フレーズテーブルの作り方

---

16. 対数線型モデルの導入
17. 句に基づく統計的機械翻訳 (SMT) における素性

---

18. デコーダ概要
19. 最小誤り率訓練

---

20. まとめ

1. オープンソースのMTシステムの例

内山将夫@NICT
mutiyama@nict.go.jp

機械翻訳手法の概要

- 人手による規則に基づく機械翻訳 (商用の機械翻訳システムのほぼ全て)
  日：主語 目的語 述語 → 英：主語 述語 目的語
  私は本を読んだ → I read a book
- コーパスに基づく機械翻訳 (現在の研究の主流)
  用例：私は本を読んだ → I read a book
  入力：私は新聞を読んだ → 出力：I read a newspaper.

コーパスベースの機械翻訳の主構成要素

- 対訳コーパス：対訳コーパスがあれば，翻訳エンジンを利用して，機械翻訳ができる．しかし，これがなければどうしようもない．→ コーパスベースの機械翻訳のボトルネック.
- 翻訳エンジン：対訳コーパスから単語や句の対訳を抽出し，それを利用して翻訳をする．現状の研究の主対象.
- 翻訳精度の評価手法：翻訳エンジンの精度を自動評価できれば，その評価を最大化するように翻訳エンジンを改良できる.

オープンソースの機械翻訳システム

http://www.statmt.org/wmt07/baseline.html

- 対訳単語等の抽出ソフトウェア
- 機械翻訳エンジン (当該ページに対訳コーパスへのポインタ)
- 翻訳精度の自動評価とそれを利用したパラメタ調整

このページだけで一応の機械翻訳ができる (学生実験レベル)
機械翻訳研究におけるベースライン.

ベースラインシステムの性能

- 解析速度：一文あたり数秒
- 翻訳精度：語順が似た言語間については，市販のシステムと同程度？
- 翻訳可能な言語：3000 万単語 (100 万文) 程度の対訳コーパスがある言語対

機械翻訳エンジンの現状と今後の課題

現状

大量の対訳コーパスが必要

対訳コーパスと異なるドメインの翻訳は苦手

主に構造が似た言語対の翻訳で実験されている

課題

少量の対訳コーパスからの学習

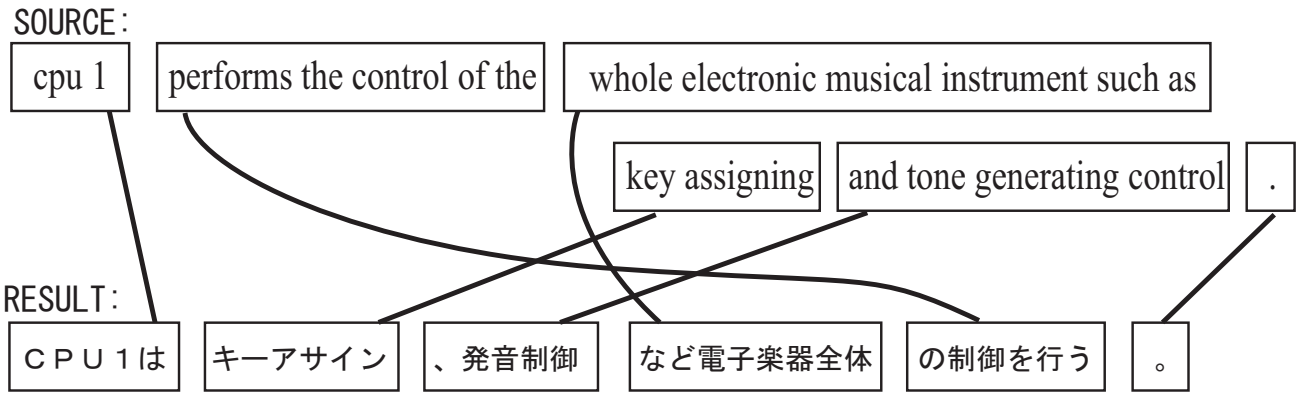
異なる分野への翻訳エンジンの適応

異なる構造の言語での翻訳エンジンの評価

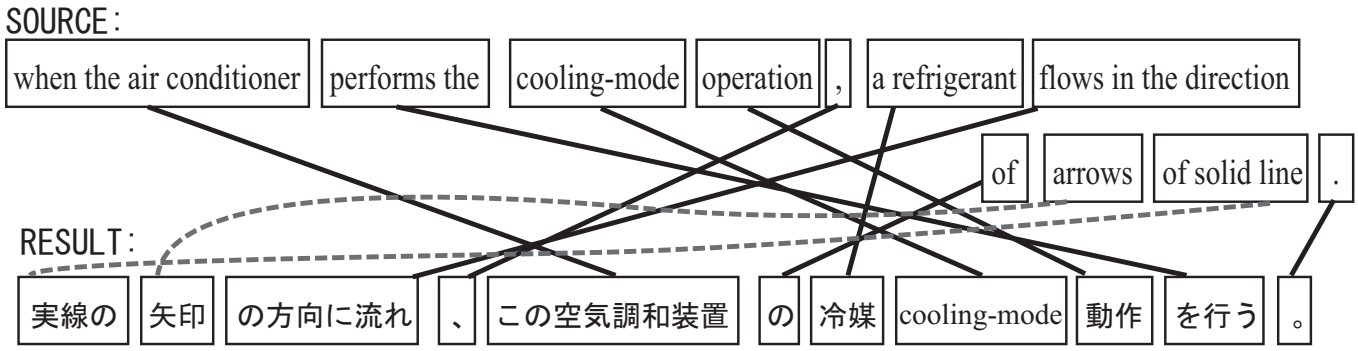
日本語のコーパスの例

コーパスベースの機械翻訳には，数十万から数百万の対訳文が現状では必要である．これに匹敵する大きさの対訳コーパスは日英では，日本と米国に同時出願された特許から構成されたものしかない．
これは，NICTが開発したものであり，700万文程度の日英の対訳文からなり，
国立情報学研究所(NII)主催のNTCIRプロジェクト
http://research.nii.ac.jp/ntcir/index-ja.html
の特許の機械翻訳タスクにおいて利用されている．
2008年12月以降に，一般にも，NIIから研究利用可能となる予定であるが，現在のところは，特許の機械翻訳タスクの参加者のみが利用可能である．

翻訳例 (1/4)



(a) 長いフレーズがヒットして正しい翻訳に成功している例（↑）



(b) フレーズはおおよそ正しいが、並び替えで失敗している例（↑）

翻訳例 (2/4)

SOURCE: consequently , the potential of internal data transmitting line io becomes higher than the potential of internal data transmitting line / io .
REFERENCE: この結果、内部データ伝達線ＩＯの電位が内部データ伝達線／ＩＯの電位よりも高くなる。
RESULT: これにより、内部データ伝達線ＩＯの電位が内部データ伝達線／ＩＯの電位よりも高くなる。

翻訳例 (3/4)

SOURCE: it is necessary to select the first and second magnetic layers 1 and 2 in order to obtain a large gmr effect .
REFERENCE: 第１及び第２磁性層１，２は高いＧＭＲ効果のために選ばれる必要がある。
RESULT: 第１及び第２の磁性層１と２を得るために、大きなＧＭＲ効果を選択する必要がある。

翻訳例 (4/4)

SOURCE: when the air conditioner performs the cooling-mode operation , a refrigerant flows in the direction of arrows of solid line .
REFERENCE: 空気調和機が冷房運転する場合、冷媒は実線矢印の向きに流れる。
RESULT: 実線の矢印の方向に流れ、この空気調和装置の冷媒 cooling-mode 動作を行う。

まとめ

ある程度の性能の計算機さえあれば，誰でも機械翻訳の研究ができるようになった.

課題

興味があれば,
http://www.statmt.org/wmt07/baseline.html
のシステムを動かしてみる．(これはそれなりに大変です)
参考情報としては,
NTCIR-7特許翻訳タスクのページ
http://if-lab.slis.tsukuba.ac.jp/fujii/ntc7patmt/index-ja.html
があります.

2. MTの性能評価についての一般的な話題

内山将夫@NICT
mutiyama@nict.go.jp

コーパスベースのMTをするときに問題になること

- MTの性能をどう測定するか？
- MTの計算モデルをどう捉えるか？
- MTの計算モデルをどう訓練するか？
- MTの訓練に必要なコーパスをどう獲得するか？

これらについて，この講義では，実例を示す.

MTの性能をどうはかるか？ — その前に —

Q: 何故 MT の性能を測定する必要があるか？

A: MTシステム A と B とを比較するとき，A と B のどちらが良いかを知るためには，A と B の性能を測定する必要がある．

Q: システムの比較はなぜ必要か？

A: より良いシステムを作るためには，システムを比較し，どちらが良いかを知る必要がある．

MTの性能は相対評価で測る

相対評価とは2つのシステムをある基準により比較すること

Q: ある基準とは何か？

A: MTを使う目的により異なる．よく使われる基準については後述する．

MTにおける絶対評価とは何か？

- よくわからないが,
- ある目的について，そのMTが
- 使えるか？使えないか？の評価だろう.
- たとえば，ケータイ翻訳で，そのケータイ翻訳だけで
- 海外旅行ができるかどうかとかが
- 絶対評価で,
- 相対評価では，2つのシステムの両方が使えなくても,
- まだ，こちらの方がましということがある.

何故相対評価をするか？

- たとえ，2つのシステムの双方が使えないにしても
- その2つのシステムを比較することにより，
- どちらのシステムの方が良いかが分かれば，
- その良い方のシステムを残して，
- それを更に改良することができるため

システムを少しずつ改良するためには，相対評価で十分である．

評価の基準は目的による

- たとえば，英文を書くときに，
- Web 上の MT システムを利用するとしたら，
- その英文作成に最適なものが良い．
- この基準は，難しいが，
- MT 訳の，ある種の「良さ」を評価するものである．

なるべく目的独立の基準が欲しい

- MTを使う目的には様々なものがあるので
- それぞれの目的ごとにMTを評価する必要があるのは
- その目的のためにMTを使う人にとっては当然である.
- しかし，MTを開発するときには,
- 全ての目的を考慮するのは無理なので,
- ある基準をもってきて，その基準を満せば,
- いろいろな目的が果せるようなものがあると良い.

課題1

- 目的独立の基準の欠点
- 目的依存の基準には，どのようなものあるか？

MT訳と参照訳が似ていれば良いのではないか?

- MTは入力文を翻訳する.
- この翻訳文が正しいものであれば，MTを使って,
- 各々の目的を達成できると思われる.
- したがって，MT訳と正しい訳(参照訳)を比較して
- それが似ているほど良いと考える.

MT訳と参照訳が似ているとはどういうことか?

- たとえば，参照訳とMT訳とを見比べて，
- 誤訳になっている個所を指摘したりする．
- あるいは，共通する単語数を数えたりする．
  → 後述する自動評価

事例：各種 Web MTシステムの比較

- 新聞記事からとった10文について,
- Q:入力英語
- A:参照訳
- として，S1,S2,S3のシステムを比較する
- 比較には，誤訳の数を数えることにする.

Q1: Europe is carrying out vigorously the Growth Initiative agreed in Edinburgh and strengthened in Copenhagen.

A: 欧州は、エディンバラにおいて合意され、コペンハーゲンにおいて強化された成長イニシアチブを精力的に実行しつつある。

S1: ヨーロッパは活発にエディンバラで同意されて、コペンハーゲンで強化されたGrowth Initiativeを実行しています。

S2: ヨーロッパは、活発に、エジンバラで同意されて、コペンハーゲンで強化されるGrowth Initiativeを実行しています。

S3: ヨーロッパは、エディンバラで同意され、コペンハーゲンで強くなった成長イニシアチブを活発に実行しています。

誤訳の数

S1: 3.「活発に」の位置.「Growth」と「Initiative」が未訳.

S2: 4.「活発に」の位置.「強化される」が誤訳.「Growth」と「Initiative」が未訳.

S3: 0.

Q2: We recognize the importance of improved market access for economic progress in Russia.

A: 我々は、ロシアの経済発展にとって、改善された市場アクセスが重要であることを認識する。

S1: 私たちはロシアに経済進歩のための立直り市況アクセスの重要性を認めます。

S2: 我々は、ロシアにおける経済進歩のために、改善された市場参入の重要性を認めます。

S3: 私たちは、ロシアで経済進歩のために改善された市場参入の重要性を認識します。

誤訳の数

S1: 3.「ロシアに」と「立直り」と「市況」が誤訳

S2: 0.

S3: 1. 「ロシアで」が誤訳

Q3: Partnerships and management assistance at corporate level can be particularly effective.

A: 法人レベルでのパートナーシップ及びマネージメント支援は、特に効果的であり得る。

S1: 法人のレベルにおけるパートナーシップと管理支援は特に有効である場合があります。

S2: 会社レベルの協力と管理援助は、特に効果的でありえます。

S3: 企業のレベルの協力および管理援助は特に有効になりえます。

誤訳の数

S1: 0

S2: 0

S3: 0

Q4: The Federal Constitutional Court decides on the question of unconstitutionality.

A: 違憲の問題については、連邦憲法裁判所が決定する。

S1: 連邦政府の Constitutional Court は違憲の問題を決めます。

S2: Federal Constitutional 法廷は、憲法違反の問題を決定します。

S3: 連邦憲法裁判所は、憲法違反の質問を決めます。

誤訳の数

S1: 3. 「Constitutional」と「Court」が未訳.「問題を」が誤訳.

S2: 3. 「Federal」と「Constitutional」が未訳.「問題を」が誤訳.

S3: 1. 「質問を」が誤訳.

Q5: The practical process of integration must begin in the economic sphere.

A: 統合の実際のプロセスは、経済分野から始めねばならない。

S1: 統合の実用的な過程は経済球で始まらなければなりません。

S2: 統合の実用的なプロセスは、経済球で始まらなければなりません。

S3: 統合の実際的なプロセスは経済球体の中で始まるに違いありません。

誤訳の数

S1: 2. 「実用的な」と「球」が誤訳

S2: 2. 「実用的な」と「球」が誤訳

S3: 2. 「球体」と「違いありません」が誤訳

Q6: Sanctions should be upheld until the conditions in the relevant Security Council resolutions are met.

A: 関連する安全保障理事会決議の諸条件が満たされるまで、制裁は維持されるべきである。

S1: 関連安全保障理事会の決議における条件が満たされるまで、制裁は是認されるべきです。

S2: 関連した安全保障理事会決議の状況が対処されるまで、制裁は支えられなければなりません。

S3: 適切な安全保障理事会の決議中の条件が満たされるまで、制裁が支持されるべきです。

誤訳の数

S1: 2. 「関連安全保障理事会」と「是認」が誤訳

S2: 3. 「状況」と「対処」と「支えられ」が誤訳

S3: 1. 「適切な」が誤訳

Q7: International terrorism is a grave threat to world peace and security.

A: 国際テロは、世界の平和と安全に対する重大な脅威だ。

S1: 国際テロは世界の平和とセキュリティへの危険な脅威です。

S2: 国際テロは、世界平和と安全に対する重大な脅威です。

S3: 国際テロは世界平和とセキュリティに対する重大な脅威です。

誤訳の数

S1: 2. 「セキュリティ」と「危険な」が誤訳

S2: 0.

S3: 1. 「セキュリティ」が誤訳

Q8: Poverty, population policy, education, health, the role of women and the well-being of children merit special attention.

A: 貧困、人口政策、教育、保健、女性の役割、及び児童の福祉は、特別の注意に値する。

S1: 貧困、人口政策、教育、健康、女性の役割、および子供の幸福は特別な注意に値します。

S2: 貧困、人口方針、教育、健康、女性の役割と子供たちの幸福は、特別な注意に値します。

S3: 欠乏、人口政策、教育、健康、女性の役割および子供の安寧は、特別の注意に値します。

誤訳の数

S1: 0.

S2: 1. 「人口方針」が誤訳

S3: 1. 「欠乏」が誤訳

Q9: Improvement of access for Russian products to international markets strongly reinforces Russian structural reform.

A: 国際市場に対するロシア産品のアクセス改善は、ロシアの構造改革を大いに強化する。

S1: 国際市場へのロシアの製品のためのアクセスの改良は強くロシアの構造改革を補強します。

S2: 国際的な市場へのロシアの製品のためのアクセスの改善は、強くロシアの構造改革を補強します。

S3: 国際市場へのロシアの製品のためのアクセスの改良は強くロシアの構造の改革を強化します。

誤訳の数

S1: 0

S2: 0

S3: 0

Q10: This also means respecting power structures established in a democratic way.

A: このことはまた民主的な形で樹立された権力構造の尊重をも意味する。

S1: また、これは、民主的な方法で確立された権力機構を尊敬するのを意味します。

S2: これも、民主主義の方向で設立される権力側を尊重することを意味します。

S3: これはさらに民主主義の方法で設立された権力機構を尊敬することを意味します。

誤訳の数

S1: 0

S2: 4. 「これも」と「方向」と「設立される」と「権力側」が誤訳

S3: 0

誤訳の集計

- S1: $3+3+0+3+2+2+2+0+0+0=15$
- S2: $4+0+0+3+2+3+0+1+0+4=17$
- S3: $0+1+0+1+2+1+1+1+0+0=7$

文単位の比較

S3 $>$ S1 $>$ S2 という傾向がありそうだ.

S1 の方が S2 より良い文の数 4 S2 の方が S1 より良い文の数 2	S1 $>$ S2
S1 の方が S3 より良い文の数 1 S3 の方が S1 より良い文の数 5	S3 $>$ S1
S2 の方が S3 より良い文の数 2 S3 の方が S2 より良い文の数 4	S3 $>$ S2

今の比較の問題点

- テスト文が少ない．10文では，十分な比較はできない．
- テスト文が恣意的である．新聞記事からとった文では，新聞記事以外の翻訳については，評価が不十分である．
- 「誤訳」の定義があいまい．なんとなく誤訳では，客観的な評価と言えない．

これらの問題点の逆を考えると

- 十分な数の文数が必要
- 恣意的でないテスト文が必要
- 「誤訳」の定義を客観的に確立する．

これらが成立してはじめて，システムの公正な比較ができる．しかし，これを全てするのは困難である．

評価に関しての最近の傾向

１つのテストセットに対して，複数の研究機関が参画する共同タスクが盛んとなっている．

- 同一データにより，異なる手法を比較できる．
- 共同のデータを利用して，よりよい評価を研究する．

良いテスト文があったとして，どれくらいの差があれば良いか？

- 比較の方法として，10文について，各文ごとに
- システムＡとＢを比べたとき，Ａが，たとえば，
- 6文について，Ｂより良かったとする．すると，
- Ａ対Ｂ＝6対4である．このとき，
- Ａの方がＢよりも良いシステムであると言ってよいか？

良くない

- なぜなら，10回中6回くらいは,
- 偶然かもしれないからである．一方,
- 10回中8回なら，これは,
- 偶然の可能性は低い．このようなことを測るために,
- 統計的検定を利用する.

Q: なぜ差があるかを比較するか？

- A: もし差があれば,
- システムを良い方向に改良するし,
- 差がなければ,
- その変更は，受け付けない.
- 差があるかどうかを知ることにより,
- システムを，その方向に変更すべきかどうかが分かる.

符号検定

- 簡単な検定方法として，符号検定がある．
- これは，システムＡとＢが同じ性能だとすると，
- ある文について，Ａが良い確率は0.5であることに基づく．
- 0.5の確率のとき，10回中6回Ａが良い確率は，

$$_{10}C_6 0.5^{10} = 0.20508$$

- である．そして，0,1,2,3,4,5回のそれぞれについては
  0.00097656，0.00097656，0.043945，0.11719，0.20508，0.24609
  である．
- したがって，Ａが0〜6回，Ｂより良い確率は，

$$\sum_{i=0}^{6} {}_{10}C_i 0.5^{10} = 0.82812$$

  である．一方
- Ａが 7〜10回良い確率は，
  1 - 0.82812 = 0.17188
  である．

確率の判断の仕方

- 6回よりもAが良い確率が
  $0.17188\ (= 17\%)$
  とかなりあるので，AとBに性能差があるとは言えない.
- 一方，8回のときには，Aが9,10回良い確率は
  $$1 - \sum_{i=0}^{8} {}_{10}C_i 0.5^{10} = 0.0107(1\%)$$
  なので,
- 10回中8回Aが良い場合には,
- そうなる確率が小さいので，統計的に有意差があると言う.

統計的検定における注意点

- AとBの性能に差があるということだけを言っていて,
- その差は，とても小さいかもしれない.
- たとえば，10000回中で，5100回，Aが良かったら,
- Aが良い割合は，0.51で,
- AとBとの差は小さいが,

$$1 - \sum_{i=0}^{5100} {}_{10000}C_i 0.5^{10000} = 0.022(2.2\%)$$

なので，有意差はある.

- つまり，有意差があるということと
- その差の大きさが十分なものかは
- 別問題である.
- また，テスト文自体が良いものかのチェックも必要である.

まとめ

- ここまでで，システムの性能を
- 相対評価で比較すること
- 評価の際には，テスト文の選び方が大切なこと
- システム間の差が有意かどうかを
- 統計的に検定できることがわかった.

課題2

- Web 上の MT システムを複数選び，その性能を比較すること

3. MTの性能をどう測定するか？ 実験の作法とMTの自動評価

内山将夫@NICT
mutiyama@nict.go.jp

実験における性能評価

MTの性能をどう測定するか？

- 実験の作法
- MTの性能の自動評価

実験の目的

MTの性能を正確に測定したい

Q: 正確とはどういうことか？

A: この実験で得られた結論が，別の実験でも成立することが，だいたい言えること．

つまり，結論が，一般化できること．

実験の作法

- 実験データを，訓練用，パラメタ調整用，テスト用に3分割する
- 訓練データで，MTモデルの基本構造を得る.
- パラメタ調整用データで，モデルのパラメタを調整する
- テストデータで，モデルの性能を確認する

モデルの性能を正しく測定するには

- 訓練，調整，テストで，データに重なりがあってはいけない．なぜなら，たとえば，
- 訓練とテストに同一のデータがあったら，単に，
- そのデータを記憶しておくだけで，高性能となる．しかし，
- 単に記憶するだけでは，未知データに対処できない．つまり，
- 一般性が低い．それにもかかわらず，
- 高性能と判断されたら，
- そのテストはおかしい．

モデルの性能を正しく測るには

色々なデータを使う必要がある
MTの場合には以下のようなものを翻訳したい

- 多言語：日英，日中，日韓,...
- 多ジャンル：新聞，論文，特許，ブログ，チャット,...

現在の技術では，ジャンルが変わっただけで，性能が大きく低下する場合が多い.
その理由：

- ジャンルが変わると，出現する単語や句が変わる

実際にはどのような実験がなされているか

- 訓練，調整，テストは必ず分ける
- できるだけたくさんの種類のデータを使う
  → しかし，
  - 実験に利用できるデータは少ないし，
  - 実験には，時間と手間が掛かるので，
  - それらを勘案して，
  - できるだけのことをする．

ともかく実験をしたとしてMTの性能をどう測るか？

2つの軸がある

- 人手評価か，自動評価か
- MT訳自体の評価か，MT訳により何ができるかによる評価か

MT訳自体の評価が評価に利用される場合が多い

考えられる理由

- MT訳自体の品質があがれば，MT訳を使ってできることの効率も上がると考えられるから
- 研究開発においては，タスクにかかわらず，MT訳を向上させることができれば，それの方が手間がかからないから

しかし，MTにより何ができるかによる評価を，もっと盛んにする必要がある．なぜなら，それこそがMTを使う理由だからである．

課題3

複数のWeb上の MT システムについて,「こういう用途なら，システムAよりもシステムBの方が良い，なぜなら,... だから」というように評価してみる．

MT訳自体の評価の方法

- 人手評価
- 自動評価

人手評価の例

- 翻訳文の流暢さ (読み易さ)
- 翻訳文の忠実度
  どのくらい原文に沿っているか？

などを各5点満点で評価するなど.

自動評価の例

- 参照用の翻訳と MT 訳との類似度を
- n-gram (n 単語連鎖) の
- 重なり具合で評価する.

参照訳と似ている訳が良い訳であるという立場である.

人手評価の良いところ

- 明確な指示を評価者に与えなくても，何らかの良さの評価ができる
- 自動評価ではできない細かな評価ができる
  文の類似性の判定には，n-gram の一致だけでなく，類似語とか，言い換えとかも考えないといけない.

人手評価における研究課題

ある評価者がある文の読み易さを中程度などと判定したとき，なにゆえ，そのような判定をしたかを調べること．たとえば，

- 間違った助詞を選択したとか
- 時制が違うとか
- 係り受けが違うとか
- 語順が違うとか

様々な要因が読み易さにはあるが，どのような要因があるかは，まだリストアップされていない．

人手評価の問題点

時間がかかる

- 1000 文の MT 訳について，その訳が良いか悪いかを判定するには
- 一週間以上かかるかもしれない.

一方,

- MT システムを開発するときには,
- すぐさま，評価結果が欲しい

自動評価ができれば，この問題は解決する.

自動評価の良いところ

すばやくできる

モデルのパラメタを調整するときに，調整用データにおけるMT訳の品質が向上する．つまり，自動評価の値が大きくなるように，パラメタを調整できる．

評価の安定性

同じMT訳と参照訳について，同一の値がでる．
→ 異なるシステムの比較が容易である．
人手評価だと，同じ人が同じ文を評価しても，異なる結果になることがある．

自動評価の悪いところ

- 自動評価では，測定できないものがある．
  たとえば，BLEU という評価尺度では，ngram の重なりしかみていないので，同一の意味でも異なる表現の場合には，間違いとされる．
- しかし，自動評価は
  - 似たタイプのシステムの比較には有用である．

まとめ

- 訓練，調整，テストにデータを分け，
  - 訓練で，モデルの基本を作り
  - 調整で，パラメタを調整し
  - テストで，テストする
- 調整にあたっては，自動評価の値が最大となるように，パラメタを調整する
- 自動評価には，限界もあるが有効なツールである．

4. 自動評価尺度 BLEU

内山将夫@NICT
mutiyama@nict.go.jp

自動評価尺度BLEU

Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. (2002) BLEU: a method for Automatic Evaluation of Machine Translation. ACL.

前提：

MT訳とプロの翻訳者による(複数の)翻訳が似ていれば似ているほど，そのMT訳は良いだろう.

自動評価に必要なもの

- (複数の)良質の翻訳
  あらかじめ用意しておく
- 似ている度合を測定する尺度 → BLEU

注：BLEUの妥当性の日本語についての現状

- これからBLEUについて,
- そのMTの評価尺度としての妥当性を検討したものは,
- 中国語–英語，アラビア語–英語が主であり,
- 中日や英日で，しっかりと検証した研究はないようである.
- これについて，我々は,
- NTCIR-7における日英特許翻訳タスクを通じて
- 調査する予定であるが,
- 今のところは，そのような調査結果はないようであるので,
- 英語を例文として利用する.

ngram の重なりによる類似度の例

1番目の良いMT訳の方が，参照訳と共通するngramが多い.

MT訳

1. $_1$It is a guide to action $_2$which $_3$ensures that the military $_4$always obyes the $_5$commands $_6$of the party.
2. It is to ensure the troops forever hearing the activity guidebook that party direct

参照訳

1. $_1$It is a guide to action that $_3$ensures that the military will forever heed Party $_5$commands.
2. It is the guiding principle $_2$which guarantees the military forces $_4$always being under the command $_6$of the party.
3. It is the practical guide for the army $_4$always to heed the directions $_6$of the party.

一般に

1. 多くの ngram を参照訳と共有するMT訳の方が
2. そうでないものよりも良い訳と言えるのではないか

欠点

1. ngram の共有は字面しかみていないので，同義語でも異なるとみなされる．また，活用を考慮していない
2. 語順があまり評価に反映されない

ngramの重なり具合の測り方の悪い例

$$\text{ngram精度} = \frac{\text{参照訳中にあるngram数}}{\text{MT訳中のngram数}}$$

不都合な例

MT: the the the the the the the

Ref1: The cat is on the mat.

Ref2: There is a cat on the mat.

MT訳はtheのみからなり，theはRef1とRef2の双方に出現しているため，上記定義だと

$$\text{1gram精度} = \frac{7}{7}$$

となる．これはおかしい．

修正されたngram精度

$$P_n = \frac{\Sigma \text{ngram ある参照訳での ngram の共有数の最大値}}{\text{MT 訳中の ngram 数}}$$

MT: the the the the the the the

Ref1: The cat is on the mat.

Ref2: There is a cat on the mat.

$P_1 = \frac{2}{7}$
$P_2 = 0$

最初の例での計算

MT1: $P_1 = \frac{17}{18} = 0.94$, $P_2 = \frac{10}{17} = 0.59$

MT2: $P_1 = \frac{8}{14} = 0.57$, $P_2 = \frac{1}{13} = 0.08$

MT訳

1. $_1$It is a guide to action $_2$which $_3$ensures that the military $_4$always obyes the $_5$commands $_6$of the party.
2. It is to ensure the troops forever hearing the activity guidebook that party direct

参照訳

1. $_1$It is a guide to action that $_3$ensures that the military will forever heed Party $_5$commands.
2. It is the guiding principle $_2$which guarantees the military forces $_4$always being under the command $_6$of the party.
3. It is the practical guide for the army $_4$always to heed the directions $_6$of the party.

複数文を翻訳したときの ngram精度

$$P_n = \frac{\Sigma_{\text{MT訳}} \Sigma_{\text{MT訳のngram}} \text{修正された共有ngram数}}{\Sigma_{\text{MT訳}} \Sigma_{\text{MT訳のngram}} \text{ngramの数}}$$

- 分母としては，全てのMT訳における全てのngramの数
- 分子は，各MT訳について，修正したngramの共有数を求めて，それを全MT訳について足したもの

修正ngram精度の統合

$$\sum_{n=1}^{N} \frac{1}{N} \log P_n$$

$P_n =$ 修正ngram精度
$N =$ ngramの最大長 (英語では4が多い)

いくつもの ngram精度を組合せることにより，数値が安定することを意図している.

長さに関しての修正ngram精度P_nの性質

参照訳よりも長いMT訳の場合

共有ngram数は，参照訳にあるngram数を越えないので，参照訳よりも長いMT訳は，P_nが小さくなる．

参照訳よりも短いMT訳の場合

短い訳のngram精度は，高くなる．これは困った．
$P_1 = 2/2,\ P_2 = 1/1$

MT訳: of the

参照訳1: It is a guide to action that ensures that the military will forever heed Party commands.

参照訳2: It is the guiding principle which guarantees the military forces always being under the command of the party.

参照訳3: It is the practical guide for the army always to heed the directions of the party.

短かすぎるMT訳へのペナルティ

MT訳の長さとコーパス中の文長(単語数)の比較

$$c = \sum_{MT\text{訳}} MT\text{訳の長さ}$$

$r = \sum_{\text{参照訳集合}}$ 参照訳中で，対応するMT訳に最近の長さ

コーパス全体で，長さを計算することにより，一文一文の長さの違いには，あまり影響されないようにする．

BP (brevity penalty)

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c \geq r \\ \exp(1 - r/c) & \text{if } c < r \end{cases}$$

- MT訳 $\geq$ 参照訳 のときには，BP $= 1$ (なにもしない)
- MT訳 $<$ 参照訳なら BP < 1 としてペナルティとする

自動評価尺度 BLEU

$$\mathrm{BLEU} = \mathrm{BP} \times \exp(\sum_{n=1}^{N} \frac{1}{N} \log P_n)$$

- 短いMT訳へのペナルティ $\times$
- ngram精度の幾何平均

BLEUにより始めて可能になること

- システムの安価で素早い比較

BLEUにより，開発→評価→開発 ... というサイクルが素早く回りはじめた．最近の統計的機械翻訳進展の原動力の一つである．
現状のMTの研究における，標準的な評価尺度である．

人手による評価とBLEUの比較

人手による評価

- 単言語グループ10人 (英語を母語とするもの)
- 2言語グループ10人 (中国語が母語)
- 各人は，中英翻訳システムが翻訳した500文の英文のうちで 50文を評価
- 各文は，5つのシステム (S1,S2,S3,H1,H2) が英語に翻訳．ただし，H1 とH2は人手による翻訳．H1は日英ともに母語ではない．H2は英語が母語.
- 評価の方法は，1(非常に悪い)〜5(非常に良い) の点を付ける.
- 単言語グループは，出力された英語の読み易さなどのみをチェック
- 2言語グループは，入力中国語と出力英語とを比較してチェック
- 各人が，各文毎に，各システムの評価をするので，システムを比較するときには，同一人の同一文におけるシステム間の評価値の差を求めて，その差が0かどうかにより，検定をする.

評価値の差の検定 (paired t-test)

- ある人u，ある文i，あるシステムsについて，評価値$r(u,i,s)$がある．
- 同じ人と文について，別のシステムs'について，評価値$r(u,i,s')$がある．
- これより，s と s' の評価値の差は $d(u,i,s,s') = r(u,i,s) - r(u,i,s')$である．
- これを全ての人と文について平均すると$m(s,s') = \Sigma_{u,i}\, d(u,i,s,s')$
- 分散は$v(s,s') = \frac{1}{n}\Sigma_{u,i}(d(u,i,s,s') - m(s,s'))^2$である．ただし，$n = \Sigma_{u,i}\, 1$.
- もし，sとs'で評価値に差がなければ，$m(s,s') = 0$なので，
- これが実際に 0 かどうかを調べるために，

$$t = \frac{m(s,s')}{\sqrt{\frac{v(s,s')}{n-1}}}$$

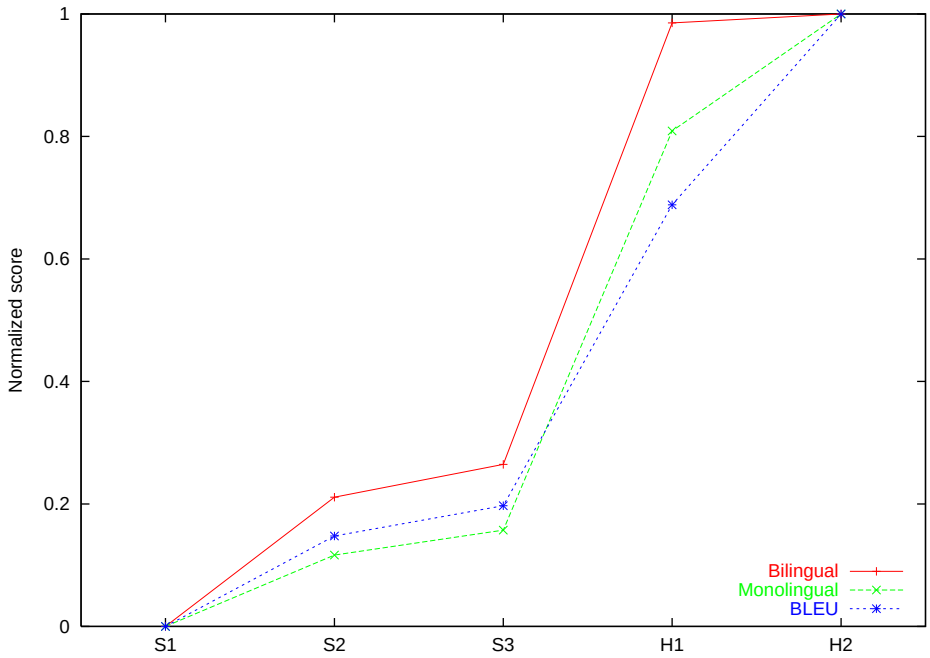
  を計算する．
- t がある値よりも大きければ，統計的に有意差がある．

BLEUおよび人手による評価値

システム	BLEU	単言語	2言語
S1	0.0527	0	0
S2	0.0829	0.326	0.551
S3	0.0930	0.44	0.691
H1	0.1934	2.265	2.574
H2	0.2571	2.8	2.612

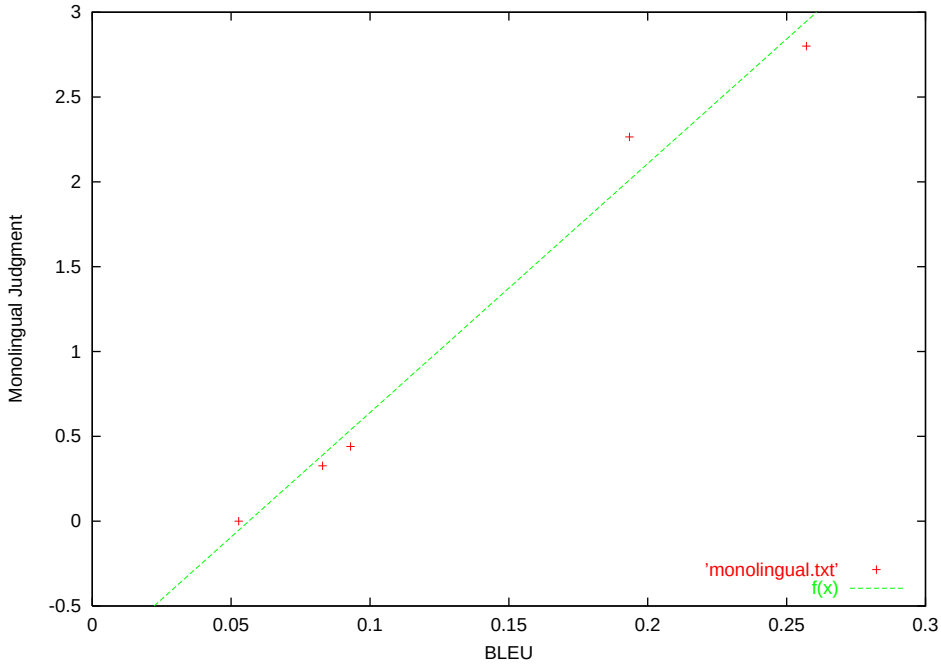
単言語グループと2言語グループにおける数値は，S1の数値を0とし，S2 は，$m(S1, S2)$ を評価値とし，$S3 = S2 + m(S3, S2)$というように，評価値の差に基づいて点数を付けた.

BLEU および人手による評価値



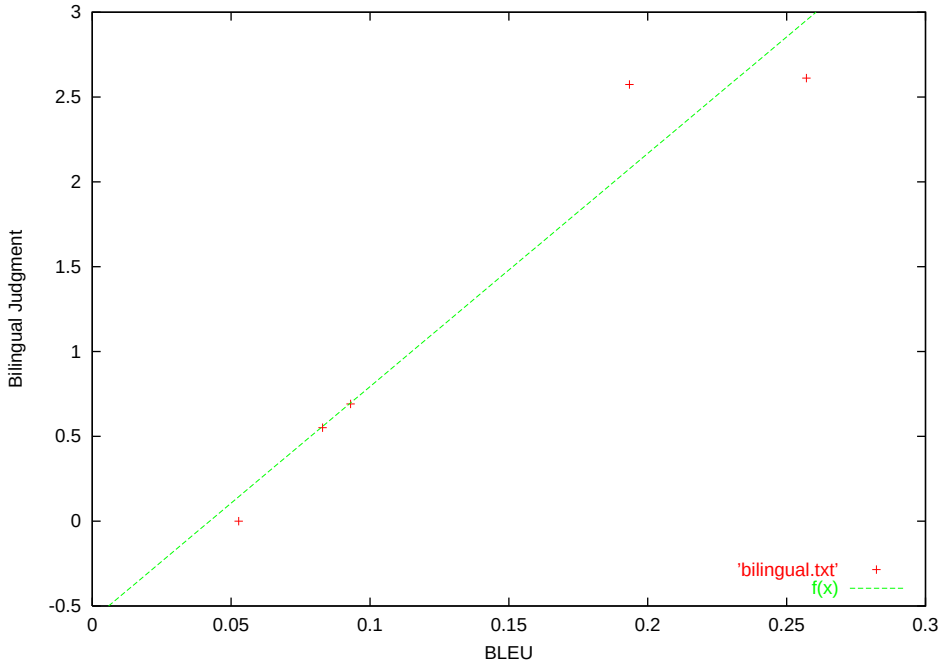
- 縦軸が評価値で，横軸がシステムである
- 0-1 に正規化したスコアを利用している
- BLEU と単言語および 2 言語の評価は似ている
- BLEU は単言語グループの評価に似ている
- S3 と H1 のような大きな差だけでなく，
- H1 と H2，S2 と S3 のような小さい差も検出可能である

BLEU と単言語グループのスコアの比較



BLEU は，単言語グループのスコアと相関が高い

BLEU と単言語グループのスコアの比較



BLEU は，2 言語グループのスコアとも相関が高い．

まとめ

- BLEU は MT 訳と参照訳との類似性を表す尺度である
- BLEU と人手評価との相関は高い

BLEU の欠点

- 意味は同じでも字面が違う ngram とマッチしない
- コーパス全体での値は求められるが文毎の値は求められない

  → どの文が上手く翻訳でき，どの文が翻訳できなかったかがわからない

しかし，現在では，ほぼ全ての研究で利用されている．

5．初歩の確率

内山将夫@NICT
mutiyama@nict.go.jp

初歩の確率

- 確率の例と用語
- 連鎖規則 (Chain Rule)
- ベイズの定理

確率の例

- 硬貨を1回投げたとき，表がでる確率
  $\rightarrow \frac{1}{2}$
- サイコロを投げたとき，1の目がでる確率
  $\rightarrow \frac{1}{6}$
- 52枚のカードから1枚をひいたとき，エースがでる確率
  $\rightarrow \frac{4}{52} = \frac{1}{13}$

確率の例 (言語モデル)

- 「今日は良い」という文字列の後に,「天気」「日」「こと」が続く確率
- 検索エンジンで「今日は良い」を検索すると，426000 ヒット
- 「今日は良い天気」は，149000
  $\rightarrow \frac{149000}{426000} = 0.35$
- 「今日は良い日」は，42200
  $\rightarrow \frac{42200}{426000} = 0.1$
- 「今日は良いこと」は，19200
  $\rightarrow \frac{19200}{426000} = 0.05$

任意の文字列 (単語列) について，確率を割当てるモデルを「言語モデル」という.

確率の例 (翻訳モデル)

「心」の訳語としては,「mind」「heart」「spirit」のどれが良く使われるか？

- 「心」と「mind」を両方含むページ → 1440000
- 「心」と「heart」を両方含むページ → 1570000
- 「心」と「spirit」を両方含むページ → 1360000

合計 = 4370000
3つが互いに排他的であるとして,

- 「心」と「mind」 → 0.33
- 「心」と「heart」 → 0.36
- 「心」と「spirit」 → 0.31

→ 日本語の単語列の翻訳確率を与えるモデルを「翻訳モデル」という

確率の例(確率による翻訳)

翻訳モデルと言語モデルを利用して訳を得る.
翻訳確率が

- 「脳」 $\rightarrow$ brain $= 1.0$, 「と」$\rightarrow$ and $= 1.0$, 「心」$\rightarrow$ mind $= 1/3$, heart $= 1/3$, spirit $= 1/3$

のとき,

翻訳モデル

- 脳と心 $\rightarrow$ (A) brain and mind $= 1/3$, (B) brain and heart $= 1/3$, (C) brain and spirit $= 1/3$

言語モデル

- (A)brain and mind のヒット数 $= 315000 \rightarrow 0.48$
- (B)brain and heart のヒット数 $= 336000 \rightarrow 0.52$
- (C)brain and spirit のヒット数 $= 800 \rightarrow 0.00$

より「翻訳モデル $\times$ 言語モデル」による推定では, (B)が優勢である.

直接数える

- (A) “brain and mind” & “脳と心” $\rightarrow 82$
- (B) “brain and heart” & “脳と心” $\rightarrow 2$
- (C) “brain and spirit” & “脳と心” $\rightarrow 0$

より (A)が優勢である. (推定と実際での食い違い)

確率の用語：条件付き確率

条件付き確率 $P(B|A)$

- Aを条件としたときのBの確率

たとえば,

- $P(\text{天気}|\text{今日は良い}) = 0.35$
- 「心」の訳語として,「mind」「heart」「spirit」しか考えないとき

  $P(\text{mind}|\text{心}) + P(\text{heart}|\text{心}) + P(\text{spirit}|\text{心}) = 1.0$

  また,
  - $P(\text{mind}|\text{心}) = 0.33$
  - $P(\text{heart}|\text{心}) = 0.36$
  - $P(\text{spirit}|\text{心}) = 0.31$

連鎖規則

$$
\begin{aligned}
&P(X_1, X_2, X_3, ..., X_N) \\
&= P(X_1) \\
&\quad \times P(X_2|X_1) \\
&\quad \times P(X_3|X_1, X_2) \\
&\quad \times P(X_4|X_1, X_2, X_3) \\
&\quad ... \\
&\quad \times P(X_N|X_1, X_2, ..., X_{N-1})
\end{aligned}
$$

たとえば,

$$
\begin{aligned}
&P(\text{今日}, \text{は}, \text{良い}, \text{天気}, \text{だ}) \\
&= P(\text{今日}) \\
&\quad \times P(\text{は}|\text{今日}) \\
&\quad \times P(\text{良い}|\text{今日}, \text{は}) \\
&\quad \times P(\text{天気}|\text{今日}, \text{は}, \text{良い}) \\
&\quad \times P(\text{だ}|\text{今日}, \text{は}, \text{良い}, \text{天気})
\end{aligned}
$$

条件付き独立

1-gram モデル

$$
\begin{aligned}
& P(X_1, X_2, X_3, ..., X_N) \\
&= P(X_1)P(X_2)P(X_3)P(X_4)\cdots P(X_N)
\end{aligned}
$$

2-gram モデル

$$
\begin{aligned}
& P(X_1, X_2, X_3, ..., X_N) \\
&= P(X_1) \\
&\quad \times P(X_2|X_1) \\
&\quad \times P(X_3|X_2) \\
&\quad \times P(X_4|X_3) \\
&\quad ... \\
&\quad \times P(X_N|X_{N-1})
\end{aligned}
$$

3-gram モデル

$$
\begin{aligned}
& P(X_1, X_2, X_3, ..., X_N) \\
&= P(X_1) \\
&\quad \times P(X_2|X_1) \\
&\quad \times P(X_3|X_1, X_2) \\
&\quad \times P(X_4|X_2, X_3) \\
&\quad ... \\
&\quad \times P(X_N|X_{N-2}, X_{N-1})
\end{aligned}
$$

ベイズの定理

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

「脳と心」が入力,「brain and mind」が翻訳候補

$$\begin{aligned} & P(\text{brain and mind}|\text{脳と心}) \\ = & \frac{P(\text{脳と心}|\text{brain and mind})P(\text{brain and mind})}{P(\text{脳と心})} \end{aligned}$$

言語モデルと翻訳モデルの組み合わせにより確率計算をする.

言語モデル

$$P(\text{brain and mind}) = P(\text{brain})P(\text{and}|\text{brain})P(\text{mind}|\text{and})$$

翻訳モデル

$$\begin{aligned} & P(\text{脳と心}|\text{brain and mind}) \\ = & P(\text{脳}|\text{brain}) \\ & \times P(\text{と}|\text{and}) \\ & \times P(\text{心}|\text{mind}) \end{aligned}$$

まとめ

- 言語現象に確率を割当てることができる
- それを利用して，翻訳確率を計算できる

6. 初歩の言語モデル

内山将夫@NICT
mutiyama@nict.go.jp

言語モデルの紹介

言語モデルというのは，任意の文字列について，それが日本語文等である確率を付与する確率モデルである．

n-gram　言語モデルとは，単語列 $w_1, w_2, \ldots, w_i$ が与えられたときに，その後に単語 x がくる確率 $P(x|w_1, w_2, \ldots, w_i)$ を，すぐ前の $n-1$ 個の単語を条件とした確率として計算する．つまり

$$P(x|w_1, w_2, \ldots, w_i) = P(x|w_{i-n+2}, w_2, \ldots, w_i)$$

以下では，単純な

- 1-gram 言語モデル
- 2-gram 言語モデル

について説明する．

1-gram 言語モデル

- 言語モデルの役割は，与えられたテキストに確率を割当ることである．
- 良くでるテキストには高い確率を割当て，そうでないものには低い確率を割当てたい．

$$
\begin{aligned}
P(\text{テキスト}) &= P(\text{単語}1, \text{単語}2, \ldots, \text{単語}m) && (1)\\
&= P(\text{単語}1)P(\text{単語}2)P(\text{単語}3)\cdots && (2)\\
&= \prod_{i=1}^{m} P(\text{単語}i) && (3)
\end{aligned}
$$

ただし，

- m はテキスト中の単語数
- 単語 i はテキストに i 番目に出現した単語

1-gram 言語モデルは，単語の確率を計算するときに文脈を考慮しない．

1-gram 言語モデルの確率推定

- 「坊っちゃん」(夏目漱石) を例に 1-gram 言語モデルを利用した確率推定の例を示す.
- 原文は ChaSen で単語に分ける.
- 全部で 2711 文あるので，先頭の 2611 文を訓練に利用して，残りの 100 文のうち 50 文をパラメタ調整，残りの 50 文をテストに使用する
- 訓練とは確率を推定することであり，テストとは推定結果を評価することである.

  (原文) 親譲り の 無鉄砲 で 小 供 の 時 から 損 ばかり し て いる 。小学校 に 居る 時分 学校 の 二 階 から 飛び降り て 一 週間 ほど 腰 を 抜かし た 事 が ある 。なぜ そんな 無闇 を し た と 聞く 人 が ある かも 知れ ぬ 。別段 深い 理由 で も ない 。新築 の 二 階 から 首 を 出し て い たら 、同級生 の 一 人 が 冗談 に 、いくら 威張っ ても 、 そこ から 飛び降りる 事 は 出来 まい 。弱虫 や ー い 。と 囃し た から で ある 。小使 に 負ぶさっ て 帰っ て 来 た 時 、 おやじ が 大きな 眼 を し て 二 階 ぐらい か ら 飛び降り て 腰 を 抜かす 奴 が ある か と 云っ た から 、 この 次 は 抜かさ ず に 飛ん で 見せ ます と 答え た 。

頻度上位の単語100語

- 延べ語数 = 55161

、 2742 。 2362 て 2092 の 2074 は 1643 が 1630 た 1599 を 1586 に 1535 と 1503 だ 1035 で 929 ない 770 から 670 し 643 も 631 な 519 おれ 451 へ 439 か 419 う 350 ん 326 」 309 「 309 ある 279 事 277 いる 245 もの 214 云う 210 人 199 する 196 たら 190 君 182 です 175 赤 172 来 171 云っ 168 い 168 よう 166 なら 166 シャツ 164 じゃ 163 そう 158 一 147 山 嵐 142 お 140 思っ 136 何 134 この 130 ば 119 てる 119 それ 119 方 114 なっ 114 いい 114 出 113 だろ 113 時 100 なる 98 まで 97 その 93 これ 92 れ 91 学校 90 ばかり 88 清 85 見 84 なり 84 や 83 聞い 81 野 78 ら 78 生徒 77 ね 77 ます 76 顔 75 さ 75 でも 74 … 74 って 73 所 72 気 70 こんな 70 校長 69 み 68 上 67 出し 67 より 67 ―― 66 二 65 行っ 65 奴 62 もし 62 うち 62 中 60 今 60 ませ 59 もん 58 なかっ 58 ちゃ 57

確率推定：最尤推定法

「、」や「。」の確率$P(\text{、})$や$P(\text{。})$を知りたい．そうするとき，最尤推定法では，

$$P(\text{テキスト}) = \prod_{i=1}^{m} P(\text{単語}i) \tag{4}$$

$$= \prod_{\text{単語}\in\text{語彙}} P(\text{単語})^{n(\text{単語})} \tag{5}$$

が最大となるように$P(\text{単語})$を定める

- 4式では，iは1からm (テキスト長) まで動くので，単語iは，i番目の単語という意味である．
- 5式では，テキスト中の同一単語の数を$n(\text{単語})$と数えて，同じ単語はひとまとめにしたものである．

最尤推定つづき

- 語彙におけるi番目の単語の確率をp_i，頻度をn_iとし
- 語彙の大きさをlとする．

$\max \Pi_{i=1}^{l} p_i^{n_i}$なる$p_i$は，テキストの確率を最尤にする．これは，$L = \Sigma_{i=1}^{l} n_i \log(p_i)$としたときに，$\max L$とすれば良い．

課題 (20分)

$\max L$ となるときにおける p_i の値を n_i を利用して表現すること．

ラグランジェの未定乗数法を使わない回答例

まず，$\Sigma_{i=1}^{l} p_i = 1$ より $p_l = 1 - \Sigma_{i=1}^{l-1} p_i$．よって

$$L = \sum_{i=1}^{l-1} n_i \log p_i + n_l \log(1 - \sum_{i=1}^{l-1} p_i) \tag{6}$$

よって，$i = 1, 2, ..., l-1$ について

$$\frac{\partial L}{\partial p_i} = \frac{n_i}{p_i} - \frac{n_l}{1 - \Sigma_{i=1}^{l-1} p_i} = 0 \tag{7}$$

つまり

$$\frac{n_i}{p_i} = \frac{n_l}{1 - \Sigma_{i=1}^{l-1} p_i} = \frac{n_l}{p_l} \tag{8}$$

よって，$K = \frac{n_l}{p_l}$ (定数) とすると，$i = 1, ..., l$ について，

$$\frac{n_i}{p_i} = K \tag{9}$$

次に，$p_i = \frac{n_i}{K}$ だから，

$$\sum_{i=1}^{l} p_i = \sum_{i=1}^{l} \frac{n_i}{K} = 1 \tag{10}$$

よって，$K = \Sigma_{i=1}^{l} n_i$．したがって，

$$p_i = \frac{n_i}{K} = \frac{n_i}{\Sigma_{i=1}^{l} n_i} \tag{11}$$

要するに，普通に出現頻度の割合を求めると，それが最尤推定確率となるということである．

ラグランジェの未定乗数法を使う回答例

$$M = \sum_{i=1}^{l} n_i \log p_i - \lambda(\sum_{i=1}^{l} p_i - 1) \tag{12}$$

とする．$i = 1, 2, ..., l$について，

$$\frac{\partial M}{\partial p_i} = \frac{n_i}{p_i} - \lambda = 0 \tag{13}$$

より，$p_i = \frac{n_i}{\lambda}$．また，制約として，

$$\frac{\partial M}{\partial \lambda} = -(\sum_{i=1}^{l} p_i - 1) = 0 \tag{14}$$

より，

$$\sum_{i=1}^{l} p_i = \sum_{i=1}^{l} \frac{n_i}{\lambda} = 1 \tag{15}$$

よって，

$$\lambda = \sum_{i=1}^{l} n_i \tag{16}$$

よって，

$$p_i = \frac{n_i}{\lambda} = \frac{n_i}{\Sigma_{i=1}^{l} n_i} \tag{17}$$

この場合には，二つの回答例であまり複雑さはかわらないが，一般には，ラグランジェの未定乗数法を利用した方が簡単である．

問題 (10分)

テキスト中の全単語の出現頻度の和 $(\Sigma_i^l n_i)$ が 55161 で,

単語	頻度
おれ	451
は	1643
蕎麦	15
が	1630
大好き	2
で	929
ある	279

という出現頻度のときに

$$P(\text{おれ，は，蕎麦，が，大好き，で，ある}) \quad (18)$$

の確率を 1-gram 言語モデルにより計算すること.

回答

単語	頻度	確率
おれ	451	$451/55161 = 0.0081760664237414 1$
は	1643	$1643/55161 = 0.029785536882942 7$
蕎麦	15	$15/55161 = 0.000271931255778539$
が	1630	$1630/55161 = 0.029549863127934 6$
大好き	2	$2/55161 = 3.625750077047 19\text{e-}05$
で	929	$929/55161 = 0.016841609107884 2$
ある	279	$279/55161 = 0.005057921357480 83$

より，確率を全て掛けて，

$$6.043909687399 61e - 18$$

最尤推定法の問題点

- 訓練テキスト中に出現しなかった単語の確率が0となる

たとえば,「坊っちゃん」で確率推定したときには$P(\text{三四郎}) = 0$となってしまう．そのため,

$$P(\text{三四郎，は，蕎麦，が，大好き，で，ある}) = 0 \quad (19)$$

となる．これは困る.
なぜ困るかというと，言語モデルの役割は,

- ある文AとBについて，AとBのうちで，よく出現しそうな文に高い確率をつけること，つまり
- $P(A) > P(B)$か$P(A) < P(B)$か$P(A) = P(B)$かを推定することだが
- 最尤推定だと，AとBに未知語(訓練データにない語)があると,
- $P(A) = P(B) = 0$となり,
- 未知語以外がどんなに違っていても同じ確率となってしまう.

未知語への対処法

- 訓練データ中にでない単語は UNK という1つの仮想的な単語と考え
- $P(\mathrm{UNK})$を推定する.

つまり,

$$P(\text{三四郎}) = P(\text{明暗}) = P(\mathrm{UNK}) \tag{20}$$

のように，でてこない単語は，1つのクラスUNKでまとめる

- この$P(UNK)$をどう推定するか

が問題である.
最尤推定だと$P(\mathrm{UNK}) = 0$となってしまうので，別の方法を使う必要がある.

未知語を含むテキストにも確率 > 0 を割当てる方法

これにはとてもたくさんの方法がある．これらの方法は，一般に，

- スムージング

と呼ばれている．
ここでは，スムージング法の中でも比較的簡単な

- 補間法

について述べる．
ngram 言語モデルを実際に利用するには，

- SRILM

等のツールキットを使うと良い．

補間法

- 複数の確率分布の重み付き平均を確率とする方法

この場合には，最尤推定による単語wの確率を$P_{ML}(w)$とすると，これにP_{ML}よりも滑らかな確率を組合せる．そのような分布として,「坊っちゃん」に出現した5507の異なり単語にUNK を加えた5508単語(K=5508とする)が一様に出現する一様分布を考えると

$$P_U = \frac{1}{K} = \frac{1}{5508} = 0.000182 \tag{21}$$

これらを足して

$$P(w) = \lambda P_{ML}(w) + (1 - \lambda)P_U \tag{22}$$

とする．ただし，$\lambda(0 \leq \lambda \leq 1)$は，重み調整用のパラメタである．

$P(\mathrm{UNK})$がどうなるか

$$
\begin{aligned}
P(\mathrm{UNK}) &= \lambda P_{ML}(\mathrm{UNK}) + (1-\lambda)P_U && (23)\\
&= \lambda \times 0 + (1-\lambda)P_U && (24)\\
&= (1-\lambda)P_U && (25)\\
&= \frac{1-\lambda}{K} && (26)
\end{aligned}
$$

これよりテキストに出現しない単語(UNK)についても確率>0が割当てられるようになった．なお，$\lambda = 1$のときには，$P(\mathrm{UNK}) = 0$である．

問題(5分)

このように定義した確率が，確率の定義を満すことを確かめること

回答例

確率の定義は, V が UNK を含む語彙として

$$\sum_{w \in V} P(w) = 1 \tag{27}$$

が成立して，かつ，全ての単語 w について

$$0 \leq P(w) \leq 1 \tag{28}$$

が成立することである.

$$\begin{aligned}
\sum_{w \in V} P(w) &= \sum_{w \in V} \{\lambda P_{ML}(w) + (1-\lambda) P_U\} && (29)\\
&= \lambda \sum_{w \in V} P_{ML}(w) + (1-\lambda) P_U \sum_{w \in V} 1 && (30)\\
&= \lambda \cdot 1 + (1-\lambda) \frac{1}{K} K && (31)\\
&= 1 && (32)
\end{aligned}$$

また，$0 \leq \lambda \leq 1$ より

$$\begin{aligned}
\lambda &\geq 0 && (33)\\
1 - \lambda &\geq 0 && (34)
\end{aligned}$$

かつ

$$\begin{aligned}
P_{ML}(w) &\geq 0 && (35)\\
P_U &> 0 && (36)
\end{aligned}$$

より,

$$P(w) = \lambda P_{ML}(w) + (1-\lambda) P_U \geq 0 \tag{37}$$

λをどう推定するか

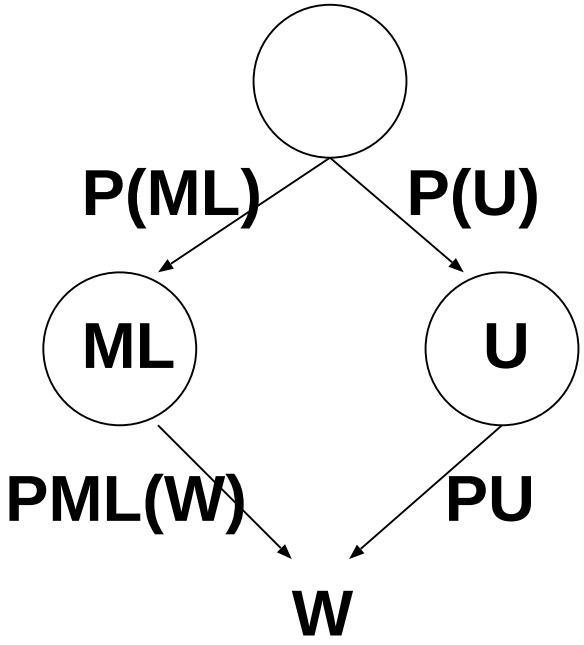
重要な点

λを調整するコーパスは，P_{ML}を推定したコーパスとは別のコーパスとすること．

理由

λの推定は最尤推定によるので，もし，P_{ML}を推定したコーパスを使うと，そこにはUNKがないので，λは 1 となってしまう．

λの推定法 (単語の生成モデル)



2つの状態 ML と U を考え，ある単語wは，MLもしくはUのどちらかからでるとすると，

$$P(w) = P_{ML}(w)P(ML) + P_U P(U) \tag{38}$$

$$P(ML) + P(U) = 1 \tag{39}$$

ところで，テキストにおけるi番目の単語t_iはMLかUのどちらかからでるので，$n(ML)$により，MLからでた単語の総頻度を表し，$n(U)$により，Uから出た単語の総頻度を表すとすると，

$$P(ML) = \frac{n(ML)}{n(ML) + n(U)} \tag{40}$$

$$P(U) = \frac{n(U)}{n(ML) + n(U)} \tag{41}$$

と推定できる．$P(ML)$がλに相当する．

ところが $n(ML)$ や $n(U)$ を直接数えることはできない．なぜなら，単語 w について，それがMLからでたかUからでたかはわからないからである．
そこで $n(ML)$ や $n(U)$ のかわりに，その期待値を使う．$n(ML)$ の期待値は

$$E(ML) = \sum_{w \in V} n(w) P(ML|w) \tag{42}$$

である．ここで，$n(w)$ は単語 w の出現頻度であり，$P(ML|w)$ は，w が与えられたときに，それがMLからでた確率である．上式は，各 w について，それが ML からどれくらいの頻度ででたかを求めて，足している．
同様に

$$E(U) = \sum_{w \in V} n(w) P(U|w) \tag{43}$$

なお，

$$P(ML|w) + P(U|w) = 1 \tag{44}$$

より

$$E(ML) + E(U) = \sum_{w \in V} n(w) = \text{全単語の頻度の和} \tag{45}$$

このとき，

$$P(ML) = \frac{E(ML)}{E(ML) + E(U)} \tag{46}$$

である．なお，$P(U) = 1 - P(ML)$ である．

よって，$P(ML|w)$を求めれば良い．ベイズの定理より

$$P(ML|w) = \frac{P(w|ML)P(ML)}{P(w)} \tag{47}$$

さて，

$P(w|ML) = P_{ML}(w) =$ 既に訓練コーパスより得られた定数 (48)

一方

$$P(ML) = \lambda \tag{49}$$

$$P(w) = \lambda P_{ML}(w) + (1-\lambda)P_U \tag{50}$$

は，今求めたい未知数λを含む．しかし，それにもかかわらず式を書くと

$$P(ML|w) = \frac{\lambda P_{ML}(w)}{\lambda P_{ML}(w) + (1-\lambda)P_U} \tag{51}$$

$$E(ML) = \sum_{w \in V} n(w)P(ML|w) \tag{52}$$

繰り返しによるλの推定

1. $\lambda = 0.5$とする
2. $$P(ML|w) = \frac{\lambda P_{ML}(w)}{\lambda P_{ML}(w) + (1-\lambda)P_U} \tag{53}$$
   $$E(ML) = \sum_{w \in V} n(w) P(ML|w) \tag{54}$$
   を計算する
3. $$\lambda = P(ML) = \frac{E(ML)}{\text{全単語の頻度の和}} \tag{55}$$
4. λが収束したら終了

この方法は，とりあえずλがわかっているものとして$E(ML)$を計算し，その結果を利用して，λを再推定するというように，少しずつλを改善する方法である．これはEM法の簡単な例である．

課題

- EM 法について調べること

EM 法は，確率統計的手法を使うときには，必須の方法ですが，EM 法自体は一般的な方法であり，それを個々の事柄に適用するためには，ここでしたように，個別の事柄にあわせて解法を開発する必要があります.
ここでは，EM 法を利用すると，未知パラメタを推定できるということを覚えておいて下さい.

例：「坊っちゃん」についてλを求める

- 2611文でP_{ML}を推定する
- 50文でλを推定する

繰り返し計算による

$\lambda, E(ML), E(U), L$

の推移をみる.

$$L = \sum_{w \in V} n(w) \log P(w) = \text{対数尤度} \tag{56}$$

$$n(w) = \text{単語} w \text{の開発データでの頻度} \tag{57}$$

$$P(w) = \lambda P_{ML}(w) + (1 - \lambda) P_u \tag{58}$$

```
# lambda=λ
# E(ML)=Pml からでた回数の期待値
# E(UNI) = Puni からでた期待値
# N=開発データの延べ単語数
# LL=対数尤度
#
ruby src/lambda.rb -n 20 botchanj/train.wfreq botchanj/test-develop.wfreq botchanj/pr
延べ単語数 = 55161
異なり単語数 = 5507
Puni = 0.000181554103122731
 1: lambda=0.7703, E(ML)=825.7380, E(UNI)=246.2620, N=1072, LL=-6693.4745
 2: lambda=0.8454, E(ML)=906.2771, E(UNI)=165.7229, N=1072, LL=-6477.1231
 3: lambda=0.8693, E(ML)=931.8799, E(UNI)=140.1201, N=1072, LL=-6451.9807
 4: lambda=0.8777, E(ML)=940.9064, E(UNI)=131.0936, N=1072, LL=-6448.6396
 5: lambda=0.8808, E(ML)=944.2322, E(UNI)=127.7678, N=1072, LL=-6448.1713
 6: lambda=0.8820, E(ML)=945.4790, E(UNI)=126.5210, N=1072, LL=-6448.1047
 7: lambda=0.8824, E(ML)=945.9495, E(UNI)=126.0505, N=1072, LL=-6448.0951
 8: lambda=0.8826, E(ML)=946.1275, E(UNI)=125.8725, N=1072, LL=-6448.0937
 9: lambda=0.8826, E(ML)=946.1949, E(UNI)=125.8051, N=1072, LL=-6448.0936
10: lambda=0.8827, E(ML)=946.2205, E(UNI)=125.7795, N=1072, LL=-6448.0935
11: lambda=0.8827, E(ML)=946.2301, E(UNI)=125.7699, N=1072, LL=-6448.0935
12: lambda=0.8827, E(ML)=946.2338, E(UNI)=125.7662, N=1072, LL=-6448.0935
13: lambda=0.8827, E(ML)=946.2352, E(UNI)=125.7648, N=1072, LL=-6448.0935
14: lambda=0.8827, E(ML)=946.2357, E(UNI)=125.7643, N=1072, LL=-6448.0935
15: lambda=0.8827, E(ML)=946.2359, E(UNI)=125.7641, N=1072, LL=-6448.0935
16: lambda=0.8827, E(ML)=946.2360, E(UNI)=125.7640, N=1072, LL=-6448.0935
17: lambda=0.8827, E(ML)=946.2360, E(UNI)=125.7640, N=1072, LL=-6448.0935
18: lambda=0.8827, E(ML)=946.2360, E(UNI)=125.7640, N=1072, LL=-6448.0935
19: lambda=0.8827, E(ML)=946.2360, E(UNI)=125.7640, N=1072, LL=-6448.0935
20: lambda=0.8827, E(ML)=946.2360, E(UNI)=125.7640, N=1072, LL=-6448.0935
```

確率の値

	最尤推定値	一様分布との補間
おれ	0.00817606642374141	0.00723817319206339
は	0.0297855368829427	0.0263124826219221
蕎麦	0.00027193l255778539	0.00026132846771908 5
が	0.029549863l279346	0.026104457435l871
大好き	3.6257500770471 9e-05	5.33032809840486e-05
で	0.0168416091078842	0.0148870992889363
ある	0.0050579213574808 3	0.0044858399521 8444
全積	6.043909687399 61e-18	4.624874917451 99e-18

UNKに確率を与えた分だけ，一様分布との補間の方が，少しずつ確率が少なくなっている．しかし，重みつき平均の結果として,「大好き」については，少し増えている．なお，$P(\mathrm{UNK}) = 2.1299406101735 3e-05$である．

2-gram 言語モデルの導入

1-gram 言語モデルでは，単語の順番と確率が無関係なので，言語のモデル化には不十分である．

P(おれ, は, 蕎麦, が, 大好き, で, ある)

$= P$(おれ)P(は)P(蕎麦)P(が)P(大好き)P(で)P(ある)

$= P$(蕎麦)P(は)P(大好き)P(が)P(で)P(ある)P(おれ)

$= P$(蕎麦, は, 大好き, が, で, ある, おれ)

2-gram 言語モデルでは，一つ前の単語を見ることにより，並び順を少し考慮する．

$$
\begin{aligned}
& P(\text{おれ}, \text{は}, \text{蕎麦}, \text{が}, \text{大好き}, \text{で}, \text{ある}) \\
&= P(\text{おれ})P(\text{は}\,|\,\text{おれ})P(\text{蕎麦}\,|\,\text{は})P(\text{が}\,|\,\text{蕎麦}) \\
&\quad P(\text{大好き}\,|\,\text{が})P(\text{で}\,|\,\text{大好き})P(\text{ある}\,|\,\text{で}) \\
&\neq P(\text{蕎麦}, \text{は}, \text{大好き}, \text{が}, \text{で}, \text{ある}, \text{おれ}) \\
&= P(\text{蕎麦})P(\text{は}\,|\,\text{蕎麦})P(\text{大好き}\,|\,\text{は})P(\text{が}\,|\,\text{大好き}) \\
&\quad P(\text{で}\,|\,\text{が})P(\text{ある}\,|\,\text{で})P(\text{おれ}|\,\text{ある})
\end{aligned}
$$

2-gram 言語モデルの求め方の例

$$P_{ML}(\text{単語}2|\,\text{単語}1) = \frac{\text{単語}1\text{の後に単語}2\text{が続く頻度}}{\text{単語}1\text{の頻度}} \tag{59}$$

例

「蕎麦」の頻度が15のとき

単語1	単語2	頻度	P(単語2|単語1)
蕎麦	屋	5	0.33
	を	4	0.27
	と	2	0.13
	粉	1	0.067
	も	1	0.067
	の	1	0.067
	が	1	0.067

最尤推定における数値例

コーパス全体での頻度 = 55161

w	v	$n(w)$	$P_{ML}(w)$	$n(v)$	$n(vw)$	$P_{ML}(w\|v)$
おれ	*	451	0.00817607			
は	おれ	1643	0.02978554	451	164	0.36363636
蕎麦	は	15	0.00027193	1643	1	0.00060864
が	蕎麦	1630	0.02954986	15	1	0.06666667
大好き	が	2	0.00003626	1630	1	0.00061350
で	大好き	929	0.01684161	2	1	0.50000000
ある	で	279	0.00505792	929	78	0.08396125

w	v	$n(w)$	$P_{ML}(w)$	$n(v)$	$n(vw)$	$P_{ML}(w\|v)$
蕎麦	*	15	0.00027193			
は	蕎麦	1643	0.02978554	15	0	0.00000000
大好き	は	2	0.00003626	1643	1	0.00060864
が	大好き	1630	0.02954986	2	0	0.00000000
で	が	929	0.01684161	1630	1	0.00061350
ある	で	279	0.00505792	929	78	0.08396125
おれ	ある	451	0.00817607	279	0	0.00000000

- 文の生成確率は，1-gram モデルでは等確率だが，2-gram モデルでは確率が異なる
- 確率 0 となる場合があるのは困る．

2-gram 言語モデルにおける補間

$$P(w|v) = \lambda_2 P_{ML}(w|v) + \lambda_1 P_{ML}(w) + \lambda_0 P_U \tag{60}$$

$$P_{ML}(w|v) = \frac{n(vw)}{n(v)} \tag{61}$$

$$P_{ML}(w) = \frac{n(w)}{\Sigma_v\, n(v)} \tag{62}$$

$$P_U = \frac{1}{|V| + 1} \tag{63}$$

$$\lambda_2, \lambda_1, \lambda_0 \geq 0 \tag{64}$$

$$\lambda_2 + \lambda_1 + \lambda_0 = 1 \tag{65}$$

問題 (15分)

テキスト $w_1, w_2, \ldots, w_N$ が与えられているとする．ただし，w_i はテキストの i 番目の単語とする．このとき，λ_0, λ_1, λ_2 の推定法を示すこと．
1-gram 言語モデルのときと同様に,

1. λ_i の初期値を設定する
2. 各 λ_i に相当する状態における期待頻度 E_i を求める．
3. E_i を利用して，λ_i を求める
4. 収束したら終了

というようにする．

λ_iの計算法

1-gram 言語モデルのときには，語彙の上で $E(ML)$ や $E(U)$ を計算したが，今回は，テキストの上で E_i を計算する．1-gram 言語モデルにおける $n(w)$ は，今回は，テキストの上で単語を動かすので，複数回出現する単語はその回数分だけ重複されてカウントされるので，1-gram 言語モデルで $n(w)$ を掛けるのと同様な効果が得られる．

1. $\lambda_2 = \lambda_1 = \lambda_0 = \frac{1}{3}$ とする
2. $$E_2 = \sum_{i=2}^{N} \frac{\lambda_2 P_{ML}(w_i|w_{i-1})}{\lambda_2 P_{ML}(w_i|w_{i-1}) + \lambda_1 P_{ML}(w_i) + \lambda_0 P_U}$$
$$E_1 = \sum_{i=2}^{N} \frac{\lambda_1 P_{ML}(w_i)}{\lambda_2 P_{ML}(w_i|w_{i-1}) + \lambda_1 P_{ML}(w_i) + \lambda_0 P_U}$$
$$E_0 = \sum_{i=2}^{N} \frac{\lambda_0 P_U}{\lambda_2 P_{ML}(w_i|w_{i-1}) + \lambda_1 P_{ML}(w_i) + \lambda_0 P_U}$$
3. $\lambda_i = \frac{E_i}{E_0+E_1+E_2}$ for $(i = 0, 1, 2)$
4. goto 2 もしくは収束したら終了

- まず初期値を設定する
- 各 w_i について，それぞれが3つの確率分布のどれからでたかの期待度数を求め，その和を各確率分布の期待度数とする．
- 期待度数の比として λ_i を求める
- 収束するまで繰り返す

数値例1

$\lambda_0 = \lambda_1 = \lambda_2 = \frac{1}{3}$のときの各単語の確率と期待度数

おれ, は, 蕎麦, が, 大好き, で, ある

	pml(w|v)	pml(w)	pu	e(w|v)	e(w)	e0
は	0.363636	0.029786	0.000182	0.923865	0.075674	0.000461
蕎麦	0.000609	0.000272	0.000182	0.573041	0.256025	0.170934
が	0.066667	0.029550	0.000182	0.691577	0.306540	0.001883
大好き	0.000613	0.000036	0.000182	0.737989	0.043615	0.218396
で	0.500000	0.016842	0.000182	0.967075	0.032574	0.000351
ある	0.083961	0.005058	0.000182	0.941262	0.056703	0.002035
sum				4.834808	0.771131	0.394061
lambda				0.805801	0.128522	0.065677

2-gramまで考慮した状態$e(w|v)$における期待頻度が他よりも大きいことがわかる.

数値例2

「坊っちゃん」のパラメタ調整用データ50文を利用してλ_iを求める.

- 2611文は，$P_U, P_{ML}(w), P_{ML}(w|v)$の推定に用いる
- 50文は，λ_iの推定に用いる

	λ2	λ1	λ0	E2	E1	E0	尤度
1:	0.5350	0.2985	0.1665	546.77	305.05	170.18	-5217.63371050
2:	0.5749	0.2842	0.1409	587.57	290.48	143.95	-5090.05722072
3:	0.5813	0.2823	0.1364	594.12	288.50	139.38	-5085.40951867
4:	0.5823	0.2823	0.1355	595.06	288.48	138.46	-5085.27277246
5:	0.5824	0.2824	0.1353	595.16	288.61	138.23	-5085.26807026
6:	0.5823	0.2825	0.1352	595.16	288.69	138.15	-5085.26776245
7:	0.5823	0.2825	0.1352	595.15	288.73	138.13	-5085.26772140
8:	0.5823	0.2825	0.1351	595.14	288.74	138.12	-5085.26771436
9:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771310
10:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771287
11:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771283
12:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
13:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
14:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
15:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
16:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
17:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
18:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
19:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282
20:	0.5823	0.2825	0.1351	595.14	288.75	138.11	-5085.26771282

尤度 $=$ $\Sigma_i \log p(w_i|w_{i-1})$が単調に増加していることがわかる.

数値例3

個別の文について各単語の確率を計算し，それを掛けて文の確率を求める．

$P(\text{おれは蕎麦が大好きである}) = 6.456 \times 10^{-14}$

w	v	$P_{ML}(w\|v)$	$P_{ML}(w)$	P_U	P
おれ	*	0.000000	0.008176	0.000182	0.002334
は	おれ	0.363636	0.029786	0.000182	0.220184
蕎麦	は	0.000609	0.000272	0.000182	0.000456
が	蕎麦	0.066667	0.029550	0.000182	0.047192
大好き	が	0.000613	0.000036	0.000182	0.000392
で	大好き	0.500000	0.016842	0.000182	0.295932
ある	で	0.083961	0.005058	0.000182	0.050344

「おれ」「は」や「大好き」「で」は，高い確率で出現する．

$P(\text{蕎麦は大好きがであるおれ}) = 1.683 \times 10^{-18}$

w	v	$P_{ML}(w\|v)$	$P_{ML}(w)$	P_U	P
蕎麦	*	0.000000	0.000272	0.000182	0.000101
は	蕎麦	0.000000	0.029786	0.000182	0.008439
大好き	は	0.000609	0.000036	0.000182	0.000389
が	大好き	0.000000	0.029550	0.000182	0.008372
で	が	0.000613	0.016842	0.000182	0.005140
ある	で	0.083961	0.005058	0.000182	0.050344
おれ	ある	0.000000	0.008176	0.000182	0.002334

- どの単語の確率も低い
- $P_{ML}(w|v) = 0$でも，$P > 0$となっている．

まとめ

- n-gram 言語モデルは，任意の文字列についての日本語らしさ (あるいは他の言語のそれらしさ) の確率を求める
- 確率の推定には，最尤推定ではなくスムージングが必要である

もっと複雑な言語モデル

- 3-gram, 4-gram, 5-gram, ... 言語モデル
- トピックを考慮した言語モデル
- 構文を利用した言語モデル

言語モデルの利用例

- 機械翻訳
- 音声認識
- 仮名漢字変換

単語対応の導入

内山将夫@NICT
mutiyama@nict.go.jp

パラレルコーパスからの情報抽出

- 単語対応 (Word alignment) の抽出 (これ)
- 句対応 (Phrase alignment) の抽出
- 翻訳規則の抽出

単語対応 (Word Alignment) とはなにか

たくさんの対訳文対が，パラレルコーパスとして与えられたとき，各対訳文対において，日本語のどの単語が，英語のどの単語に対応しているかを同定する．
単語対応は，そもそも良く定義できない問題である．つまり，対訳文が与えられたとき，どの単語とどの単語が対応するかは，人間がみても良くわからないのが普通である．
しかし，今のところ，単語対応を求めることは，コーパスベースの翻訳にとって，本質的な問題である．

単語対応の例

- 「今日」「は」「良い」「天気」「だ 」
- 「It」「is」「fine」「today」

において

- 「今日」と「today」は対応する.
- 「良い」「天気」と「fine」は対応する.

その他の「は」「だ」や「It」「is」は，どう判断したら良いだろうか？

課題

適当な対訳文10文について，単語対応を求めること.

単語対応の確率モデル (Brown et al. 1993)

$P(\mathbf{f}|\mathbf{e})$ = フランス語の文$\mathbf{f}$が英語の文$\mathbf{e}$から生成される確率

この $P(\mathbf{f}|\mathbf{e})$ を，英単語と仏単語の対応確率から求めたい．まず，$\mathbf{e}$ と $\mathbf{f}$ は，それぞれ l,m 個の単語からなる．

$$\mathbf{e} = e_1 e_2 \ldots e_i \ldots e_l = e_1^l \tag{1}$$

$$\mathbf{f} = f_1 f_2 \ldots f_j \ldots f_m = f_1^m \tag{2}$$

次に，各仏単語 f_j が，ただ一つの英単語に対応するとして，その英単語の位置を a_j とする．

$$\mathbf{a} = a_1 a_2 \ldots a_j \ldots a_m = a_1^m \tag{3}$$

このとき，f_j に対応する英単語は，e_{a_j} である．もし，f_j に対応する英単語がない場合には，$a_j = 0$ とする．この f_j は，NULL単語から生成されたと考える．したがって，$a_j \in \{0, 1, \ldots, l\}$ である．ある対応,

	f_1	f_2	f_3	f_4	f_5
NULL					x
e_1		x			
e_2	x		x		
e_3				x	

において，$f_1 \to e_2$, $f_2 \to e_1$, $f_3 \to e_2$, $f_4 \to e_3$, $f_5 \to \mathrm{NULL}(e_0)$ より $a_1 = 2$, $a_2 = 1$, $a_3 = 2$, $a_4 = 3$, $a_5 = 0$ である．

ある生成モデル (IBM-Model 1,2 の原型)

$$
\begin{aligned}
P(\mathbf{f}|\mathbf{e}) &= \sum_{\mathbf{a}} P(\mathbf{f},\mathbf{a}|\mathbf{e}) \\
P(\mathbf{f},\mathbf{a}|\mathbf{e}) &= \text{仏文}\mathbf{f}\text{と対応}\mathbf{a}\text{が英文}\mathbf{e}\text{から生成される確率} \\
&= P(m|\mathbf{e}) \prod_{j=1}^{m} P(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \\
&\quad P(f_j|a_1^{j}, f_1^{j-1}, m, \mathbf{e})
\end{aligned}
$$

$$P(m|\mathbf{e}) = \text{英文}\mathbf{e}\text{が}m\text{単語の仏文になる確率}$$

$P(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) =$ 仏文が f_1^{j-1} まで生成され，それぞれが，a_1^{j-1} の位置の英単語につながっているとき，j 番目の仏語が a_j 番目の英単語につながる確率

$P(f_j|a_1^{j}, f_1^{j-1}, m, \mathbf{e}) =$ 上述の条件に加えて，j 番目の仏語が a_j 番目の単語につながるときに，j 番目の仏単語が f_j である確率

これは，確率の式を厳密に展開したものであることに注意する．上述の式を簡単化したものがIBM Model 1,2 である．別の形の式展開をするとIBM Model 3,4,5となる．

8. IBM Model-1 の式の説明

内山将夫@NICT
mutiyama@nict.go.jp

IBM-Mode1の式の説明のまえに

Q:
何のために確率モデルを利用するか
A:

- 単語対応の確率や
- 単語対応自体を求めるため

単語対応の確率：
「お金」は「cash」と「money」のどちらに訳されることが多いか？
単語対応自体を求めるため：
「今日」「は」「良い」「天気」「だ」と「It」「is」「fine」「today」では,「今日」はどの単語に対応するか？

単語対応自体を求めるにはどうするか

条件：
英文$\mathbf{e}$と仏文$\mathbf{f}$が与えられていて，単語対応$\mathbf{a}$について，$P(\mathbf{f}, \mathbf{a}|\mathbf{e})$が計算できるとする
求めるもの$\hat{\mathbf{a}}$:

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e})$$

$\hat{\mathbf{a}}$は，$P(\mathbf{f}, \mathbf{a}|\mathbf{e})$を最大化するアラインメントである．これは最も確率が高いアラインメントである．
$\rightarrow$ アラインメントの確率をモデル化することにより，そのモデルの下での最適アラインメントを求めることができる．

単語対応の確率を求めにはどうするか

- 求めたいパラメタをθとして，明示的に式に導入すると

$$P(\mathbf{f}|\mathbf{e},\theta)$$

により，θをパラメタとしたときの$\mathbf{f}$の確率がわかる

- パラレルコーパスを$\mathbf{d}=\{\langle\mathbf{f},\mathbf{e}\rangle\}$とすると

$$L(\theta|\mathbf{d})=\log\prod_{\langle\mathbf{f},\mathbf{e}\rangle\in\mathbf{d}}P(\mathbf{f}|\mathbf{e},\theta)$$

によりパラメタθの下での，コーパス全体での$\mathbf{f}$の確率がわかる．このとき

$$\hat{\theta}=\arg\max_{\theta}L(\theta|\mathbf{d})$$

なる$\hat{\theta}$をみつけると，コーパスを最尤で生成するθが求まる．

ここまでのまとめ

確率でモデル化することにより

- 単語対応(アラインメント)が計算できる
- 単語対応確率等のパラメタが計算できる

→ 大きな利点

難しいところ

- モデルの正当性の評価が難しい
- 数式で色々なことを表現しないといけないので，あまり複雑なことは表現できない
- モデルの良さは，実際にデータに適用してみないとわからない
- 良いアイディアと思っても上手くいくかはわからない

IBM model-1の式展開

$$P(\mathbf{f},\mathbf{a}|\mathbf{e}) = P(m|\mathbf{e}) \prod_{j=1}^{m} P(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) P(f_j|a_1^{j}, f_1^{j-1}, m, \mathbf{e}) \quad (1)$$

$$\begin{aligned} P(m|\mathbf{e}) &= \text{仏文の長さ(単語数)が}m\text{である確率} \\ &= \text{ある定数}\epsilon \end{aligned}$$

$$\begin{aligned} P(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) &= j\text{番目の仏単語がつながるのが} \\ &\quad a_j\text{番目の英単語の確率} \\ &= \text{どの場所にも同確率} \\ &= \frac{1}{l+1} \end{aligned}$$

$$\begin{aligned} &P(f_j|a_1^{j}, f_1^{j-1}, m, \mathbf{e}) \\ &= j\text{番目の仏単語が}f_j\text{の確率} \\ &= f_j\text{と}e_{a_j}\text{のみで決まる} \\ &= t(f_j|e_{a_j}) \end{aligned}$$

以上より

$$P(\mathbf{f},\mathbf{a}|\mathbf{e}) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(f_j|e_{a_j}) \quad (2)$$

アラインメント$\mathbf{a}$について和をとる

$$
\begin{aligned}
& P(\mathbf{f}|\mathbf{e}) \\
&= \sum_{\mathbf{a}} P(\mathbf{f},\mathbf{a}|\mathbf{e}) \\
&= \sum_{\mathbf{a}} \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(f_j|e_{a_j}) \\
&= \frac{\epsilon}{(l+1)^m} \sum_{\mathbf{a}} \prod_{j=1}^{m} t(f_j|e_{a_j}) \\
&= \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \qquad (3)
\end{aligned}
$$

$\mathbf{a} = a_1 ... a_m$におけるa_jは仏語f_jが英単語e_{a_j}に継がることを示す．また，$\mathbf{e} = e_1 ... e_l$で，かつ，$e_0 = \mathrm{NULL}$である． そのため，$0 \leq a_j \leq l$である．更に，$1 \leq j \leq m$なので，上式の ように変形される．

パラメタ推定

3 式において推定すべきパラメタは $t(f|e)$ のみである．これには前述のように，

$$\max P(\mathbf{f}|\mathbf{e})$$

となる $t(\cdot)$ を推定すれば良い．
制約として，各英単語 e について

$$\sum_f t(f|e) = 1$$

である．つまり，各英単語 e について，それに対応する f の確率の和は 1 である．

ラグランジェの未定乗数法により極値を求める

3式と制約より

$$h(t,\lambda) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) - \sum_e \lambda_e (\sum_f t(f|e) - 1) \tag{4}$$

第2項が制約部分であり，これはλ_eで偏微分をすることにより，

$$\frac{\partial h}{\partial \lambda_e} = -(\sum_f t(f|e) - 1) = 0 \tag{5}$$

となる．つまり，λ_eによる偏微分の結果が0であることにより，極値におけるパラメタ値が制約を満すようになる．

一方，$t(f|e)$による偏微分においては，4式の第2項は，

$$\frac{\partial}{\partial t(f|e)} - \sum_e \lambda_e (\sum_f t(f|e) - 1) = -\lambda_e \tag{6}$$

4式の第1項は，まず，

$$\prod_{j=1}^{m} t(f_j|e_{a_j})$$

を偏微分する．

$$
\begin{aligned}
&\frac{\partial}{\partial t(f|e)} \prod_{j=1}^{m} t(f_j|e_{a_j}) \\
&= \left(\prod_{\substack{1\le k\le m \\ f_k\neq f \vee e_{a_k}\neq e}} t(f_k|e_{a_k}) \right) \frac{\partial}{\partial t(f|e)} t(f|e)^{\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})} \\
&= \left(\prod_{\substack{1\le k\le m \\ f_k\neq f \vee e_{a_k}\neq e}} t(f_k|e_{a_k}) \right) n(e,f) t(f|e)^{n(e,f)-1} \\
&= n(e,f) t(f|e)^{-1} \left(\prod_{\substack{1\le k\le m \\ f_k\neq f \vee e_{a_k}\neq e}} t(f_k|e_{a_k}) \right) t(f|e)^{n(e,f)} \\
&= n(e,f) t(f|e)^{-1} \prod_{1\le k\le m} t(f_k|e_{a_k}) \qquad (7)
\end{aligned}
$$

ただし，

$$
\begin{aligned}
n(e,f) &= \sum_{j=1}^{m} \delta(f,f_j)\delta(e,e_{a_j}) \\
&= \text{仏単語が}\, f_j \,\text{で英単語が}\, a_j \,\text{の単語対応の数}
\end{aligned}
$$

- 1行目では，$t(f|e)$の項を取り出す
- 2行目では，$t(f|e)$で偏微分する
- 3,4 行目では，$t(f|e)^{-1}$ を括り出すことにより，$\Pi_{1\le k\le m}\, t(f_k|e_{a_k})$を再導入する

以上より，

$$\frac{\partial}{\partial t(f|e)}h(t,\lambda)$$
$$= \frac{\epsilon}{(l+1)^m}\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\left(\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})\right)t(f|e)^{-1}$$
$$\prod_{1\leq k\leq m}t(f_k|e_{a_k})-\lambda_e \tag{8}$$

これを0とおくと

$$t(f|e) = \lambda_e^{-1}\frac{\epsilon}{(l+1)^m}$$
$$\times \sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\left(\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})\right)\prod_{1\leq k\leq m}t(f_k|e_{a_k}) \tag{9}$$

まず，λ_eについて説明すると，これは4式において，$\Sigma_f\, t(f|e) = 1$となるように導入した変数である．この制約を満すには，$t(f|e) = \lambda_e^{-1}A(f|e)$としたとき，$\lambda_e = \Sigma_f\, A(f|e)$とすると，$\Sigma_f\, t(f|e) = 1$となる．つまり，$\lambda_e$は，単なる正規化定数である．

これまでのまとめ

- 9式により，$t(f|e)$ を表現した
- この式を利用すると $t(f|e)$ が求まると思われる
- しかし右辺にも $t(\cdot)$ がでてくる

EM 法により $t(\cdot)$ を求める

1. まず $t(\cdot)$ に適当な初期値を定める
2. 次に，その値を利用して，9式により $t(\cdot)$ を計算する
3. 2 を繰り返す

このようにして，$t(e|f)$ を求める道筋が整った．次は，9式を簡単化していく．

9式の再掲

$$t(f|e) = \lambda_e^{-1} \frac{\epsilon}{(l+1)^m} \times \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \left(\sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j}) \right) \prod_{1 \le k \le m} t(f_k|e_{a_k})$$

3式の再掲

$$P(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j})$$

これらより

$$t(f|e) = \lambda_e^{-1} \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e}) \sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j}) \qquad (10)$$

これを更に概念的に簡単化するために,

$$C(f|e; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{f}, \mathbf{e}) \sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j}) \qquad (11)$$

を定義する.

この式の$\Sigma_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j})$は，アラインメント$\mathbf{a}$において$f$と$e$が継がっている回数である．また，$P(\mathbf{a}|\mathbf{f}, \mathbf{e})$は，アラインメント$\mathbf{a}$の確率である．よって，$C(f|e; \mathbf{f}, \mathbf{e})$は，$f$と$e$が対応する回数の期待値である.

これから確認するように，$t(f|e)$は，fとeの対応の期待度数$C(f|e, \mathbf{f}, \mathbf{e})$を利用した割合により表現される.

$$
\begin{aligned}
t(f|e) &= \lambda_e^{-1} \sum_{\mathbf{a}} P(\mathbf{f},\mathbf{a}|\mathbf{e}) \sum_{j=1}^{m} \delta(f,f_j)\delta(e,e_{a_j}) \\
&= \lambda_e^{-1} \sum_{\mathbf{a}} P(\mathbf{f}|\mathbf{e}) P(\mathbf{a}|\mathbf{f},\mathbf{e}) \sum_{j=1}^{m} \delta(f,f_j)\delta(e,e_{a_j}) \\
&= \lambda_e^{-1} P(\mathbf{f}|\mathbf{e}) \sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{f},\mathbf{e}) \sum_{j=1}^{m} \delta(f,f_j)\delta(e,e_{a_j}) \\
&= \lambda_e^{-1} P(\mathbf{f}|\mathbf{e}) C(f|e;\mathbf{f},\mathbf{e}) \\
&= \lambda_e'^{-1} C(f|e;\mathbf{f},\mathbf{e}) \qquad (12)
\end{aligned}
$$

$\lambda'_e = \lambda_e P(\mathbf{f}|\mathbf{e})^{-1}$は正規化定数．$\lambda'_e = \Sigma_f\, C(f|e;\mathbf{f},\mathbf{e})$とすれば，$\Sigma_f\, t(f|e) = 1$となる．

上記は，英文$\mathbf{e}$と仏文$\mathbf{f}$が一文だけしかない場合である．コーパス全体では，s番目の対訳文を$\mathbf{e}^{(s)}$と$\mathbf{f}^{(s)}$で表すと

$$
t(f|e) = \lambda_e'^{-1} \sum_{s} C(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)})
$$

である．これは，期待度数をコーパス全体で測定している．

これまでのまとめ

$$t(f|e) = \lambda'^{-1}_e \sum_s C(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}) \quad (13)$$

$$\lambda'_e = \sum_f \sum_s C(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}) \quad (14)$$

$$C(f|e;\mathbf{f},\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{f},\mathbf{e}) \sum_{j=1}^{m} \delta(f,f_j)\delta(e,e_{a_j}) \quad (15)$$

上式を利用した推定アルゴリズム

1. 全ての f と e について，$t(f|e)$ の初期値を選ぶ.
2. コーパス中の文 s について $C(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)})$ を計算する.
3. 各 e について，λ'_e を計算する
4. 各 f について，$t(f|e)$ を計算する
5. goto 2 or exit

残された問題

$$
\begin{aligned}
& C(f|e;\mathbf{f},\mathbf{e}) \\
&= \sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{f},\mathbf{e}) \sum_{j=1}^{m} \delta(f,f_j)\delta(e,e_{a_j}) \\
&= \lambda_e P(\mathbf{f}|\mathbf{e})^{-1} t(f|e) \qquad (\text{式}12\text{参照}) \qquad (16)
\end{aligned}
$$

をどう計算するか？
4式に戻ると

$$
h(t,\lambda) = \frac{\epsilon}{(l+1)^m} \underline{\sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j})} \\
- \sum_e \lambda_e (\sum_f t(f|e) - 1)
$$

下線の部分で$(l+1)^m m$回の計算が必要になる．この部分を簡単化する必要がある．

$$\sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) = \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i)$$

$(l+1)^m m$回の計算量が$m(l+1)$に減少した.
例により，この変形の正しさを確認する.
$m=3, l=1$のとき，$t(f_j|e_i) = t_{ji}$として,

$$\begin{aligned} \text{左辺} &= \sum_{a_1=0}^{1} \sum_{a_2=0}^{1} \sum_{a_3=0}^{1} \prod_{j=1}^{3} t(f_j|e_{a_j}) \\ &= t_{10}t_{20}t_{30} + t_{11}t_{20}t_{30} + \cdots + t_{11}t_{21}t_{30} + t_{11}t_{21}t_{31} \end{aligned}$$

$$\begin{aligned} \text{右辺} &= \prod_{j=1}^{3} \sum_{i=0}^{1} t(f_j|e_i) \\ &= (t_{10}+t_{11})(t_{20}+t_{21})(t_{30}+t_{31}) \end{aligned}$$

4式を簡単化する

$$h(t,\lambda) \;=\; \frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m}\sum_{i=0}^{l} t(f_j|e_i) - \sum_e \lambda_e(\sum_f t(f|e) - 1)$$

第2項の$t(f|e)$による偏微分は$-\lambda_e$.

$$n_f \;=\; \sum_{j=1}^{m} \delta(f, f_j) = (f = f_j)\text{なる}\, f_j\text{の数}$$

$$n_e \;=\; \sum_{i=0}^{l} \delta(e, e_i) = (e = e_i)\text{なる}\, e_i\text{の数}$$

とすると

$$\begin{aligned}
&\prod_{j=1}^{m}\sum_{i=0}^{l} t(f_j|e_i) \\
&= \left(\prod_{\substack{1\le j\le m\\ f_j\neq f}}\sum_{i=0}^{l} t(f_j|e_i)\right)\left(\sum_{i=0}^{l} t(f|e_i)\right)^{n_f} \\
&= \left(\prod_{\substack{1\le j\le m\\ f_j\neq f}}\sum_{i=0}^{l} t(f_j|e_i)\right)\left(\sum_{\substack{0\le i\le l\\ e_i\neq e}} t(f|e_i) + n_e t(f|e)\right)^{n_f} \qquad (17)
\end{aligned}$$

だから

$$\begin{aligned}
&\frac{\partial\text{第}2\text{項}}{\partial t(f|e)} \\
&= n_f n_e \left(\sum_{\substack{0\le i\le l\\ e_i\neq e}} t(f|e_i) + n_e t(f|e)\right)^{n_f-1} \\
&= n_f n_e \left(\sum_{0\le i\le l} t(f|e_i)\right)^{n_f-1} \\
&= \frac{n_f n_e}{\Sigma_{0\le i\le l}\, t(f|e_i)} \left(\sum_{0\le i\le l} t(f|e_i)\right)^{n_f} \qquad (18)
\end{aligned}$$

以上より，

$$\frac{\partial}{\partial t(f|e)} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) = \frac{n_f n_e}{\Sigma_{0\leq i\leq l}\, t(f|e_i)} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \tag{19}$$

よって，

$$\begin{aligned} &\frac{\partial h(t,\lambda)}{\partial t(f|e)} \\ &= \frac{n_f n_e}{\Sigma_{i=0}^{l}\, t(f|e_i)} \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) - \lambda_e \\ &= \frac{n_f n_e}{\Sigma_{i=0}^{l}\, t(f|e_i)} P(\mathbf{f}|\mathbf{e}) - \lambda_e \end{aligned} \tag{20}$$

よって，極値では $\frac{\partial h(t,\lambda)}{\partial t(f|e)} = 0$ より

$$\lambda_e P(\mathbf{f}|\mathbf{e})^{-1} = \frac{n_f n_e}{\Sigma_{i=0}^{l}\, t(f|e_i)}$$

よって，16式より

$$\begin{aligned} &C(f|e;\mathbf{f},\mathbf{e}) \\ &= \lambda_e P(\mathbf{f}|\mathbf{e})^{-1} t(f|e) \\ &= \frac{t(f|e)}{\Sigma_{i=0}^{l}\, t(f|e_i)} n_f n_e \end{aligned} \tag{21}$$

ここで，$\frac{t(f|e)}{\Sigma_{i=0}^{l}\, t(f|e_i)}$ は，f と e という対応に与えられる重みであり，$n_f n_e$ は，そのような対応の数である．

核となる式

$$C(f|e;\mathbf{f},\mathbf{e}) = \frac{t(f|e)}{\sum_{i=0}^{l} t(f|e_i)} n_f n_e \tag{22}$$

$$\lambda_e = \sum_f \sum_s C(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}) \tag{23}$$

$$t(f|e) = \lambda_e^{-1} \sum_s C(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}) \tag{24}$$

$t(\cdot)$推定のアルゴリズム

1. $t(f|e)$の初期値を設定する
2. 各文sについて$C(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)})$を計算する．
3. 各eについて，λ_eを計算する．
4. 各fについて，$t(f|e)$を計算する．
5. goto 2 or exit.

課題

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. (1993) The Mathematics of Statistical Machine Translation: Parameter Estimation.
を読んでみる．

9. IBM Model-1 の動作例

内山将夫@NICT
mutiyama@nict.go.jp

小さいコーパス

- 「彼」「の」「絵 」と「his」「painting」
- 「彼」「の」「コレクション」と「his」「collection」
- 「絵」「の」「コレクション」と「painting」「collection」

初期値

f が日本語単語で，eが英単語に相当する．
$t(f|e) = \frac{1}{\text{日本語単語の異なり語数}} = \frac{1}{4}$
$c(f|e) = \Sigma_f \Sigma_s C(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)}) = 0$
と初期値を設定する．

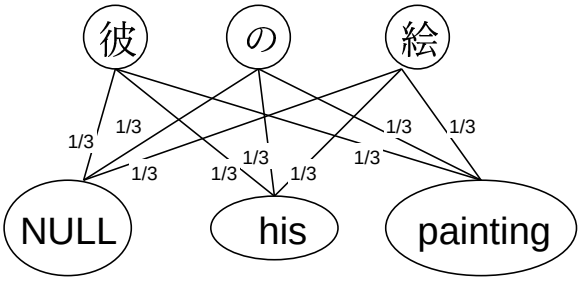
$t(f|e)/c(f|e)$

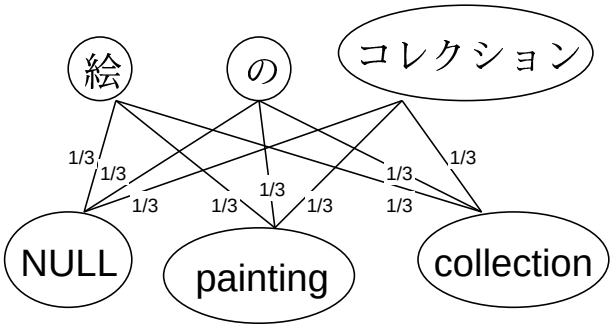
	彼	の	絵	コレクション	確率の計
NULL	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	1
his	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	1
painting	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	1
collection	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	$\frac{1}{4}/0$	1

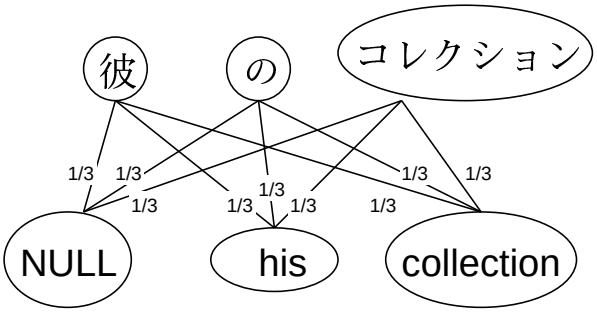
1回目の計算

全てのエッジの重みが$\frac{1}{3}$である．たとえば，
「彼」「の」「絵」と「NULL」「his」「painting」において

$$\frac{t(\text{絵}|\text{painting})}{t(\text{絵}|\text{NULL})+t(\text{絵}|\text{his})+t(\text{絵}|\text{painting})} = \frac{1/4}{1/4+1/4+1/4} = \frac{1}{3}$$







$C(f|e)$の集計 / $t(f|e)$の再推定

$$C(f|e) = f\text{と}e\text{をつなぐエッジの重みの総和}$$

$$t(f|e) = \frac{C(f|e)}{\Sigma_f C(f|e)} = \frac{C(f|e)}{C(e)}$$

「NULL」と「彼」に注目すると，このペアは2回でたので

$$C(\text{彼}|\text{NULL}) = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

一方

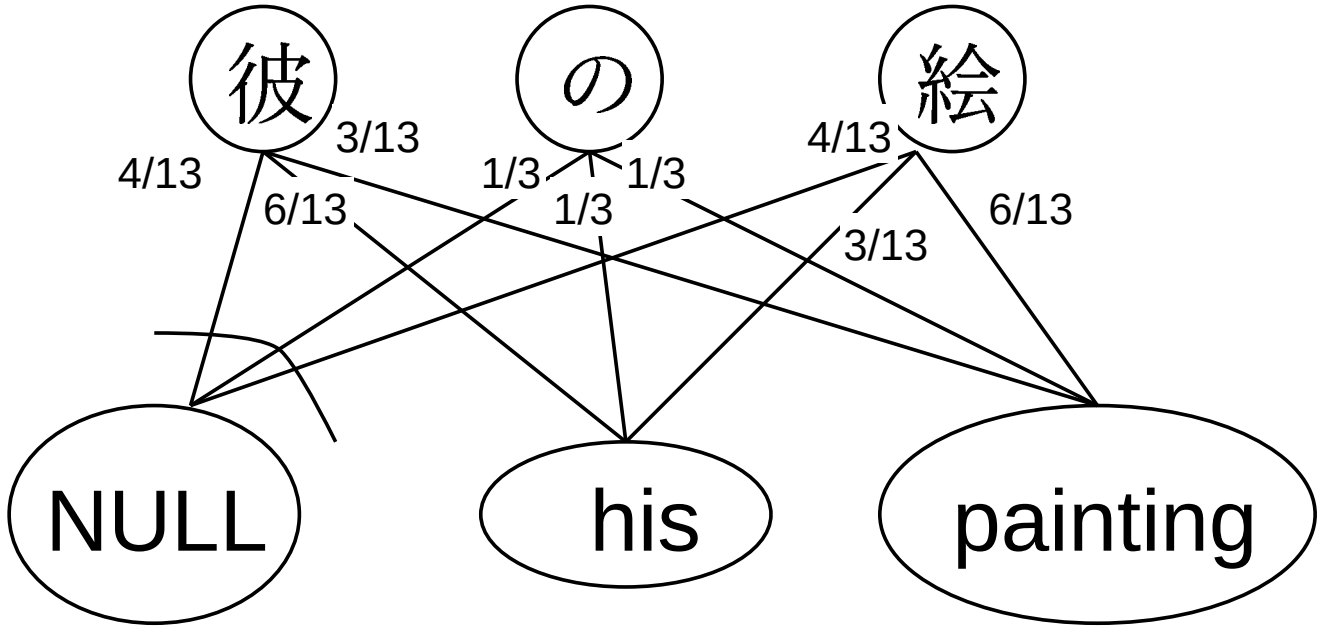
$$C(\text{NULL}) = \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} = \frac{9}{3}$$

よって，

$$t(\text{彼}|\text{NULL}) = \frac{C(\text{彼}|\text{NULL})}{C(\text{NULL})} = \frac{\frac{2}{3}}{\frac{9}{3}} = \frac{2}{9}$$

	彼	の	絵	コレ	$C(e)$
NULL	$\frac{1}{3} + \frac{1}{3}/\frac{2}{9}$	$\frac{1}{3} + \frac{1}{3} + \frac{1}{3}/\frac{3}{9}$	$\frac{1}{3} + \frac{1}{3}/\frac{2}{9}$	$\frac{1}{3} + \frac{1}{3}/\frac{2}{9}$	$\frac{9}{3}$
his	$\frac{1}{3} + \frac{1}{3}/\frac{2}{6}$	$\frac{1}{3} + \frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{6}{3}$
paint.	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3} + \frac{1}{3}/\frac{2}{6}$	$\frac{1}{3} + \frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{6}{3}$
coll.	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3} + \frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3} + \frac{1}{3}/\frac{2}{6}$	$\frac{6}{3}$

2回目の計算：文1

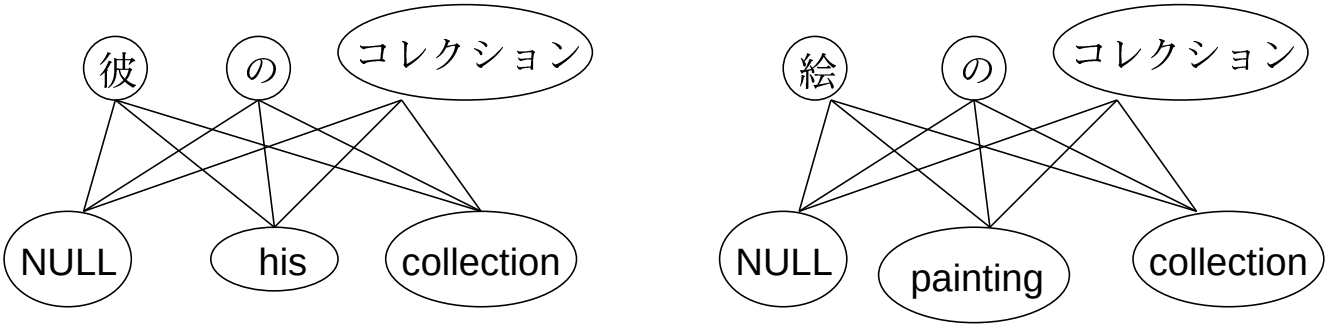


$$\frac{t(\text{彼}|\text{NULL})}{t(\text{彼}|\text{NULL})+t(\text{彼}|\text{his})+t(\text{彼}|\text{painting})} = \frac{\frac{2}{9}}{\frac{2}{9}+\frac{2}{6}+\frac{1}{6}} = \frac{4}{13}$$

$$\frac{t(\text{の}|\text{NULL})}{t(\text{の}|\text{NULL})+t(\text{の}|\text{his})+t(\text{の}|\text{painting})} = \frac{\frac{3}{9}}{\frac{3}{9}+\frac{2}{6}+\frac{2}{6}} = \frac{1}{3}$$

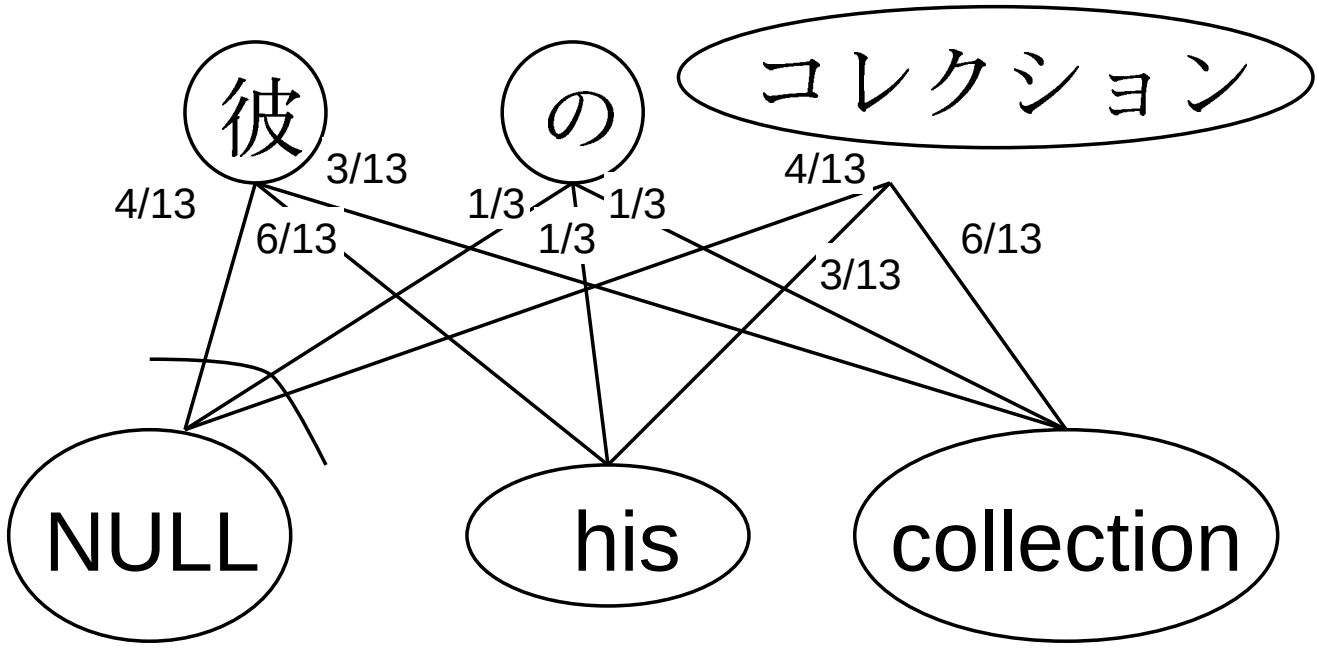
$$\frac{t(\text{絵}|\text{NULL})}{t(\text{絵}|\text{NULL})+t(\text{絵}|\text{his})+t(\text{絵}|\text{painting})} = \frac{\frac{2}{9}}{\frac{2}{9}+\frac{1}{6}+\frac{2}{6}} = \frac{4}{13}$$

問題 (15分)



上記2文における各エッジの重みを求めること

2回目の計算：文2

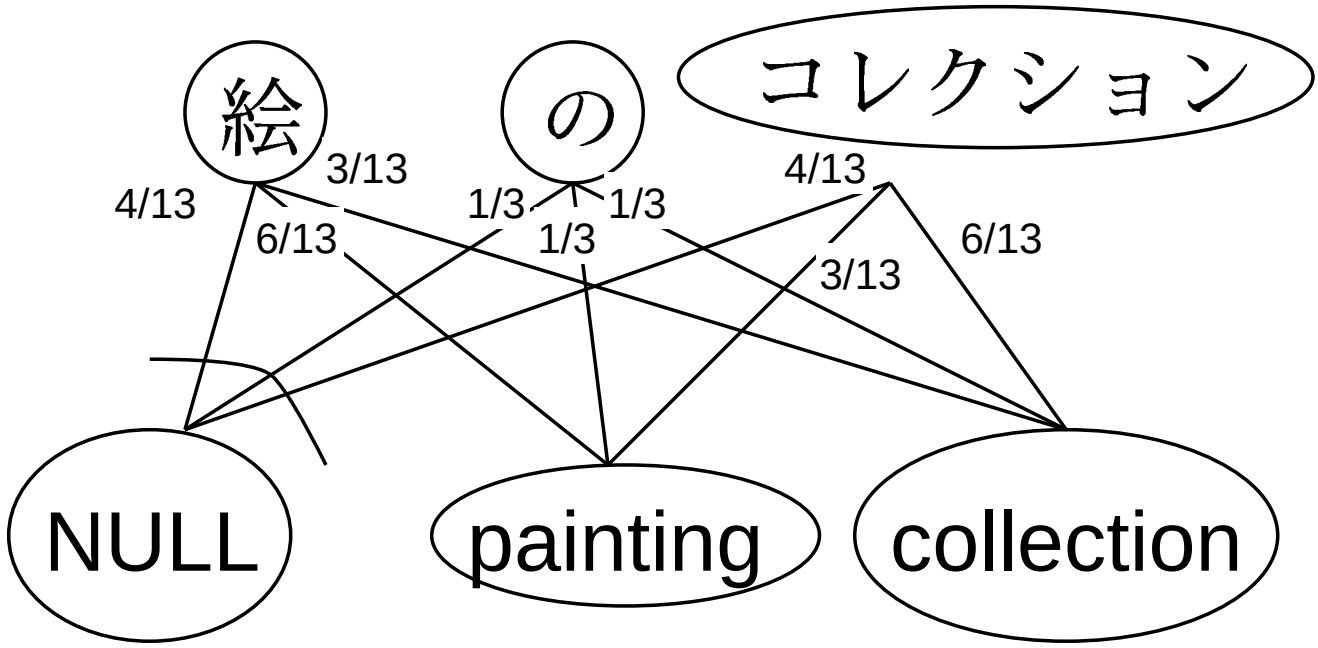


$$\frac{t(\text{彼}|\text{NULL})}{t(\text{彼}|\text{NULL})+t(\text{彼}|\text{his})+t(\text{彼}|\text{collection})} = \frac{\frac{2}{9}}{\frac{2}{9}+\frac{2}{6}+\frac{1}{6}} = \frac{4}{13}$$

$$\frac{t(\text{の}|\text{NULL})}{t(\text{の}|\text{NULL})+t(\text{の}|\text{his})+t(\text{の}|\text{collection})} = \frac{\frac{3}{9}}{\frac{3}{9}+\frac{2}{6}+\frac{2}{6}} = \frac{1}{3}$$

$$\frac{t(\text{コレクション}|\text{NULL})}{t(\text{コ}|\text{NULL})+t(\text{コ}|\text{his})+t(\text{コ}|\text{collection})} = \frac{\frac{2}{9}}{\frac{2}{9}+\frac{1}{6}+\frac{2}{6}} = \frac{4}{13}$$

2回目の計算：文3



$$\frac{t(\text{絵}|\text{NULL})}{t(\text{絵}|\text{NULL})+t(\text{絵}|\text{painting})+t(\text{絵}|\text{collection})} = \frac{\frac{2}{9}}{\frac{2}{9}+\frac{2}{6}+\frac{1}{6}} = \frac{4}{13}$$

$$\frac{t(\text{の}|\text{NULL})}{t(\text{の}|\text{NULL})+t(\text{の}|\text{painting})+t(\text{の}|\text{collection})} = \frac{\frac{3}{9}}{\frac{3}{9}+\frac{2}{6}+\frac{2}{6}} = \frac{1}{3}$$

$$\frac{t(\text{コレクション}|\text{NULL})}{t(\text{コ}|\text{NULL})+t(\text{コ}|\text{painting})+t(\text{コ}|\text{collection})} = \frac{\frac{2}{9}}{\frac{2}{9}+\frac{1}{6}+\frac{2}{6}} = \frac{4}{13}$$

問題 (15分)

前述の$C(f|e)$の集計 / $t(f|e)$の再推定における

	彼	の	絵	コレ	$C(e)$
NULL	$\frac{1}{3}+\frac{1}{3}/\frac{2}{9}$	$\frac{1}{3}+\frac{1}{3}+\frac{1}{3}/\frac{3}{9}$	$\frac{1}{3}+\frac{1}{3}/\frac{2}{9}$	$\frac{1}{3}+\frac{1}{3}/\frac{2}{9}$	$\frac{9}{3}$
his	$\frac{1}{3}+\frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}+\frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{6}{3}$
paint.	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3}+\frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}+\frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{6}{3}$
coll.	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3}+\frac{1}{3}/\frac{2}{6}$	$\frac{1}{3}/\frac{1}{6}$	$\frac{1}{3}+\frac{1}{3}/\frac{2}{6}$	$\frac{6}{3}$

の表を更新すること

$C(f|e)$の集計 / $t(f|e)$の再推定

$$C(f|e) = f\text{と}e\text{をつなぐエッジの重みの総和}$$

$$t(f|e) = \frac{C(f|e)}{\Sigma_f C(f|e)} = \frac{C(f|e)}{C(e)}$$

「NULL」と「彼」に注目すると，

$$C(\text{彼}|\text{NULL}) = \frac{4}{13} + \frac{4}{13} = \frac{8}{13}$$

$$C(\text{NULL}) = \frac{4}{13} + \frac{4}{13} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{4}{13} + \frac{4}{13} + \frac{4}{13} + \frac{4}{13} = \frac{37}{13}$$

よって，

$$t(\text{彼}|\text{NULL}) = \frac{C(\text{彼}|\text{NULL})}{C(\text{NULL})} = \frac{\frac{8}{13}}{\frac{37}{13}} = \frac{8}{37}$$

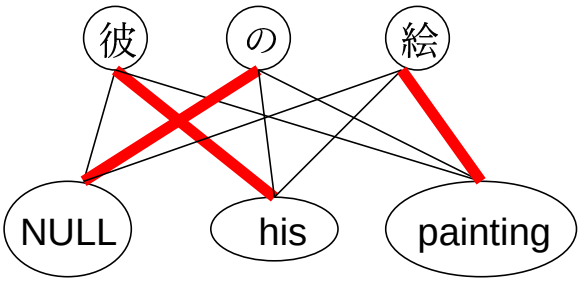
	彼	の	絵	コレ	$C(e)$
NULL	$\frac{4}{13} + \frac{4}{13}/\frac{8}{37}$	$\frac{1}{3} + \frac{1}{3} + \frac{1}{3}/\frac{13}{37}$	$\frac{4}{13} + \frac{4}{13}/\frac{8}{37}$	$\frac{4}{13} + \frac{4}{13}/\frac{8}{37}$	$\frac{37}{13}$
his	$\frac{6}{13} + \frac{6}{13}/\frac{36}{80}$	$\frac{1}{3} + \frac{1}{3}/\frac{26}{80}$	$\frac{3}{13}/\frac{9}{80}$	$\frac{3}{13}/\frac{9}{80}$	$\frac{80}{39}$
paint.	$\frac{3}{13}/\frac{9}{80}$	$\frac{1}{3} + \frac{1}{3}/\frac{26}{80}$	$\frac{6}{13} + \frac{6}{13}/\frac{36}{80}$	$\frac{3}{13}/\frac{9}{80}$	$\frac{80}{39}$
coll.	$\frac{3}{13}/\frac{9}{80}$	$\frac{1}{3} + \frac{1}{3}/\frac{26}{80}$	$\frac{3}{13}/\frac{9}{80}$	$\frac{6}{13} + \frac{6}{13}/\frac{36}{80}$	$\frac{80}{39}$

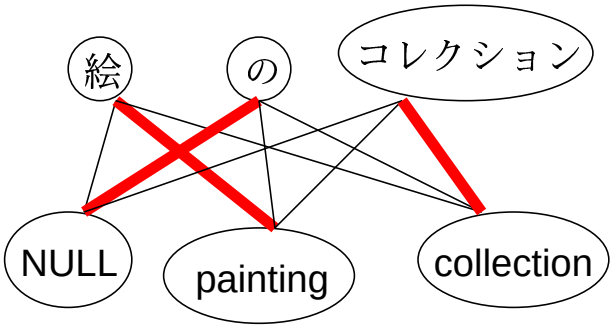
赤いセルに確率が集中していくことがわかる．

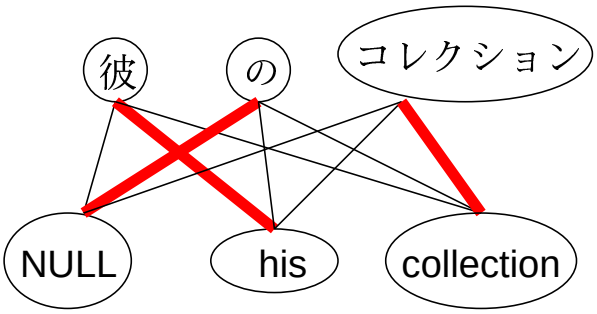
	彼	の	絵	コレ	$C(e)$
NULL	.615/.216	1.0/.351	.615/.216	.615/.216	2.846
his	.923/.45	.667/.325	.231/.113	.231/.113	2.051
paint.	.231/.113	.667/.325	.923/.45	.231/.113	2.051
coll.	.231/.113	.667/.325	.231/.113	.923/.45	2.051

単語アラインメント

確率最大の$t(f|e)$について赤くする．







日英対訳文対応付けデータ(内山・高橋2003)での例

約10万文の小説等のコーパスからIBM Model-1で得られた対訳確率が0.5以上でかつ無作為に抽出した単語対の例

英日方向

経済/economic, ワルツ/waltz, 宮殿/palace, 音楽/music, フリー/free, ウワア/wow, ：/:, 1999/1999, US/us, ファーディナンド/ferdinand, 〃/., 百万/million, 砂地/sand, ７．/7., ほこり/dust, 伯父/uncle, 霜/frost, ブリ/gabriel, 停留所/stop, ポール/paul, ウマ/horse, オックスフォード/oxford, 両方/both, サー/sir, 財産/property, ハドソン/hudson, 薄い/thin, エウリュピュロス/eurypylus, ベルリン/berlin, ２０３．/the, ?/?, 高等/higher, 音節/syllables, イグネイシャス/ignatius, ピナー/pinner, 日/day, ヶ月/months, １０１．/the, 銅/copper, コンロイ/conroy, ハウス/house, ベーアマン/behrman, ほぼ/almost, 唇/lips, 新/new, 主題/subject, 祭壇/altar, エドワード/edward, 太陽/sun, Software/software, 現象/phenomena, 空中/air, 友情/friendship, ガ/gallaher, 戦う/fight, 君たち/you, 気付/hearthrug, 三月/march, 被曝/exposure, ５９．/mischievous, ヘンリー/henry, ワトスン/watson, マシン/machine, イタリア/italy,)/), エンジニアリング/engineering, 1986/1986, ２２９．/the, 美しい/beautiful, ドードー/dodo, 損失/loss, 呑み/gasped, シカゴ/chicago, Web/web, 砂漠/desert, 地主/squire, 朝食/breakfast, バーベキュー/barbecue, 1509/1509, デザイナー/designer, 下っ/down, アクセス/access, Law/law, 青/blue, 星/stars, ジョンストン/johnston, 蓮/lotus, ズン/thump, アンリエッタ/henrietta, エルサレム/jerusalem, 風の音/wind

日英方向

nose/鼻, amen/アーメン, ..7-1/., network/ネットワーク, dorothy/ドロシー, .ix/., joe/ジョー, winter/冬, animals/動物, perrault/ペロー, window/窓, julia/ジュリア, church/教会, smoke/煙, reasons/理由, trousers/ズボン, patten/パッテン, armour/武具, tooth/歯, passepartout/パスパルトゥー, troy/トロイア, es/., era/時代, freedom/自由, flag/旗, dublin/ダブリン, churches/教会, table/テーブル, eagle/ワシ, darwin/ダーウィン, poem/詩, daisy/デイジー, daughter/娘, shoulders/肩, twyford/トワイフォード, 2/2, branches/枝, maimie/マイミー, “/「, usecbc/UseCBC, master/主人, jim/ジム, uh/はい, purposes/目的, 20/20, straits/海峡, importation/輸入, endfor/EndFor, film/映画, fiddle-stick/ばかばかしい, 1997/1997, fox/キツネ, history/歴史, 12/12, flower/花, german/ドイツ, years/年, eveline/エヴリン, religion/宗教, register/レジスタ, skirt/スカート, poole/プール, lisp/LISP, slowly/ゆっくり, tink/ティンク, mit/MIT, kiotsukete/キヲツケテ, woods/森, puck/パック, head/頭, lysander/ライサンダー, french/フランス, insurance/保険, reproduction/複製, ..3-1/., horses/馬, ..5-1/., .41/., israeli/イスラエル, christianity/キリスト教, priam/プリアモス, ohio/オハイオ, intellect/知性, shoulder/肩, wall/壁, buildings/建物, truths/真理, rights/権利, gnu/GNU, gentleman/紳士, 16/16, program/プログラム

アラインメントの例

X: 日英方向
Y: 英日方向
両方向で選ばれた対応：

“と「，angelと天使，likeとよう

	「	天使	の	よう	だ	ね	」
“	XY				X		
he							
looks		Y				X	
just		Y					
like				XY			
an		Y	X				
angle		XY					
“							Y

まとめ

- IBM Model-1 の実行例を観察した
- 対訳単語の抽出や単語対応がとれることをみた
- Model-1はそれほど強力なモデルではない

課題

興味があれば，`http://www2.nict.go.jp/x/x161/members/mutiyama/software.html`にあるIBM Model-1 のプログラムを実行してみる

10. 現状で利用可能なパラレル(対訳)コーパス

内山将夫@NICT
mutiyama@nict.go.jp

これまでのまとめ

コーパスベースの機械翻訳により
対訳コーパスから自動的に
翻訳機を作ることができることを述べ,
その手始めとして，IBM Model-1 を説明した.

ここでやること

ここでは，機械翻訳自体の話題は少し休んで,
コーパスベースの機械翻訳に必要な
対訳コーパスを
どう手に入れたら良いかを話す.

ここでの話題

- 対訳コーパスの重要性について
- 現状で利用可能な対訳コーパス
- 対訳コーパスを自動で作る方法
- 機械翻訳以外における対訳コーパスの利用

対訳コーパスとはなにか

複数言語について，特に，
意味内容がほぼ等しいと考えられる文について
対応関係が付いているコーパスを
パラレルコーパスあるいは対訳コーパスと呼ぶ

対訳コーパスの例

オオカミと仔ヒツジ
The Wolf and the Lamb
ある日のこと、オオカミは、群とはぐれて迷子になった仔ヒツジと出会った。
WOLF, meeting with a Lamb astray from the fold,
オオカミは、仔ヒツジを食ってやろうと思ったが、牙を剥いて襲いかかるばかりが能じゃない。
resolved not to lay violent hands on him,
何か上手い理由をでっち上げて手に入れてやろうと考えた。
but to find some plea to justify to the Lamb the Wolf's right to eat him.
そこで、オオカミはこんなことを言った。
He thus addressed him:
「昨年お前は、俺様にひどい悪口を言ったな！」
"Sirrah, last year you grossly insulted me."
仔ヒツジは、声を震わせて答えた。
「誓って真実を申しますが、私はその頃、まだ生まれていませんでした。」
"Indeed," bleated the Lamb in a mournful tone of voice, "I was not then born."
するとオオカミが言った。
Then said the Wolf,
「お前は、俺様の牧草を食べただろう！」
"You feed in my pasture."

対訳コーパスの重要性

- 対訳コーパスは，コーパスベースの機械翻訳にとって，前提条件である
- 歴史的には，
  - まず，対訳コーパスが自動構築され
  - つぎに，それを利用して，統計的機械翻訳が研究された

対訳コーパスはなぜ重要か

統計的機械翻訳では，入力文$\mathbf{f}$について

$$\hat{\mathbf{e}} = \arg\max_{e} P(\mathbf{e}|\mathbf{f})$$

なる$\hat{\mathbf{e}}$を出力するが，
この確率の推定には，手本となる対訳コーパスが必要だからである．

対訳コーパス構築の困難性

しかし，利用可能な対訳コーパスは少ない (後述)
その理由は，対訳コーパスの構築には，様々なことを解決しないといけないからである．

対訳コーパス構築における障害

- 原文および翻訳文の獲得
  - 原文および翻訳文には，著作権保持者がいる．
  - この著作権保持者の許可を得ないと
  - その文章は対訳コーパスに採用できない
  - たくさんの著作権保持者と交渉するのは，時間も費用もかかる
- 高価

  たとえ対訳コーパスがあったとしても

  それらは，しばしば，高価である．

研究利用可能なコーパスの例

- Linguistic Data Consortium のコーパス
  年会費2500ドル(30万弱)を払うことにより,
  - 中国語 - 英語
  - アラビア語 - 英語
  - フランス語 - 英語

  の対訳コーパスを入手可能である．各言語対について100万文以上ある．
- Europarl コーパス
  欧州諸語についてのコーパスが無償で利用可能である．各言語対について数十万の規模である．
- NICTで公開しているコーパス (無償)
  日本語と英語の対訳コーパス．30万文程度である．
- NTCIRで利用可能なコーパス (無償)
  NICTで開発した日米特許対訳コーパス180万文を利用して，特許翻訳タスクが実施されている．

NICTで無償公開している対訳コーパス

- 読売新聞とThe Daily Yomiuriの対訳文18万文
  http://www2.nict.go.jp/x/x161/members/mutiyama/jea/index-ja.html
- 小説など160作品の対訳約10万文
  http://www2.nict.go.jp/x/x161/members/mutiyama/align/index.html

日英新聞記事対応付けデータ

「読売新聞」と「The Daily Yomiuri」とで互いに翻訳関係にあるような文を自動的に対応付けたデータ

- 1対1文対応：15万，1対n文対応：3万
- 日英2言語コーパスとしては世界最初のある程度の規模のコーパス
- 研究教育目的に利用可能

文対応の精度は 98% 程度

サンプル

- 欧州は、エディンバラにおいて合意され、コペンハーゲンにおいて強化された成長イニシアチブを精力的に実行しつつある。 Europe is carrying out vigorously the Growth Initiative agreed in Edinburgh and strengthened in Copenhagen.
- 我々は、ロシアの経済発展にとって、改善された市場アクセスが重要であることを認識する。We recognize the importance of improved market access for economic progress in Russia.
- 法人レベルでのパートナーシップ及びマネージメント支援は、特に効果的であり得る。Partnerships and management assistance at corporate level can be particularly effective.

検索例

low profile

「Japan has kept a low profile」「日本は目立った行動を控えていた」

dispose of

「dispose of bad assets」「不良資産の処理」

データの特徴

- 新聞記事
- 高品質な実例

欲しい表現全てを網羅するほど大きくはないが，頻出表現は網羅できる．
→ コーパスに基づく英語教育に利用

日英対訳文対応付けデータ

再配布可能な日英の作品について，対訳文対応を1文単位で付けたデータ

- Project Gutenbergや青空文庫やプロジェクト杉田玄白などの作品について
- 160作品を公開

Project Gutenberg

Project Gutenberg, abbreviated as PG, is a volunteer effort to digitize, archive and distribute cultural works. Founded in 1971 by Michael Hart, it is the oldest digital library.[1] Most of the items in its collection are the full texts of public domain books. The project tries to make these as free as possible, in long-lasting, open formats that can be used on almost any computer. As of August 2007, Project Gutenberg claimed over 22,000 items in its collection. Project Gutenberg is affiliated with many projects that are independent organizations which share the same ideals, and have been given permission to use the Project Gutenberg trademark.

From Wikipedia, the free encyclopedia

青空文庫

青空文庫は，利用に対価を求めない，インターネット電子図書館です．著作権の消滅した作品と，「自由に読んでもらってかまわない」とされたものを，テキストとXHTML (一部は HTML) 形式でそろえいています．
「青空文庫早わかり」より

プロジェクト杉田玄白

プロジェクト杉田玄白というのは、いろんな文章を勝手に翻訳して公開しちゃうプロジェクトなのだ。プロジェクトグーテンベルグや、青空文庫の翻訳版だと思って欲しい。日本は翻訳文化だといわれるけれど、それならいろんな翻訳が手軽に入手できるようにすることで、もっともっと文化的な発展ができるようになるだろう。

作品の一部

「80日間世界一周」「DESのクラック：暗号研究と盗聴政策、チップ設計の秘密」「RMS スウェーデン王立工科大学講演」「「年」の話」「『群集心理』」「あらし」「ひと，場所，もの，そして，アイデア」「わがままな大男」「アッシャー家の崩壊」「アモンティリャードの酒樽」「アラビー」「アルセーヌ・ルパンの逮捕」「イソップ寓話集」「エヴリン　「ダブリンの人々」より」「オズの魔法使い」「カール・マルクス Interview」「カウンターパーツ」「キリストにならいて」「クリスマス・カロル」「グレイト・ギャツビー」「グロリア・スコット号」「ケンジントン公園のピーターパン」

データの特徴

- 小説やエッセイ

楽しみのために読むことができる.

デモ

11. パラレルコーパスを利用した検索と対数尤度比検定による対訳抽出

内山将夫@NICT
mutiyama@nict.go.jp

パラレルコーパスの英語学習への利用

- 辞書を調べるような感覚でパラレルコーパスを調べる
- 自由に利用できる対訳コーパス検索システム
  「日英対応付けコーパスの検索」

  http://www.kotonoba.net/~snj/cgi-bin/text-search/text-search.cgi

日英対応付けコーパスの検索サイト

- 35万文の対訳コーパス
  - 読売新聞とThe Daily Yomiuri 18万文
  - ロイター日英 7万文
  - 小説等 10万文
- 月間で200〜300人程度が使っている模様

検索される語の例

- 「バランス.*取」「Maintaining a balance between the technocrats' liberalism and economic nationalism, and curbing corruption among the nationalists was of keyimportance to the stability and economic development of Indonesian society.」「こうした構図にあっては、テクノクラートの自由主義と経済ナショナリズムの＜＜＜バランスがうまく取＞＞＞れ、しかもナショナリストに伴いがちな腐敗・汚職がそこそこコントロールされていることが、社会の安定と経済の発展には重要であった。」
- 「んじゃない」「you must not touch her.」「さわる＜＜＜んじゃない＞＞＞。」「But, I think the public might be convinced if everyone nominated Mr. Nakasone for the premiership.」「本当は中曽根さんを皆で担いだら、国民も納得する＜＜＜んじゃない＞＞＞か。」

英語教育への利用

- 日本大学中條清美先生

  http://www5d.biglobe.ne.jp/~chujo/resorce.html

  - パラレルコーパスを利用した語彙指導タスク集
  - パラレルコーパスを利用した文法指導タスク集1
  - パラレルコーパスを利用した文法指導タスク集2

タスクの例

1. 「decline」の日本語訳で多いものをあげてみよう.「下落」「減少」「衰退」「低下」「落ち込み」
2. 「efficiency」の日本語訳で多いものをあげてみよう.「効率」「燃費」「能率」
3. 「製品」にあたる英語で特に多いものは何ですか.「product」「products」
4. どんな製品がありますか.「(...) products」となるものを3つ見つけて，日本語訳もつけましょう.「foreign products (外国製品)」「industrial products (工業製品)」「oil products (石油製品)」「steel products (鉄鋼製品)」
5. 「commercial (...)」という用例を2つ見つけて日本語訳をつけよう.「commercial areas (商業地)」「commercial bank (都市銀行)」
6. 「inventory (...)」という用例を2つ見つけて日本語訳をつけよう.「inventory adjustment (在庫調整)」「inventory index (在庫指数)」
7. 「(...) access」という用例を2つ見つけて日本語訳をつけよう.「free access (自由なアクセス)」「Internet access (インターネットアクセス)」

対訳候補の抽出

対訳コーパスを便利に使うには,
「efficiency」の日本語訳で多いものをあげてみよう
という質問に対して,
「効率」「燃費」「能率」
という回答が，素早く分かることが必要である.
そのために,

1. decline と特に良く共起する日本語単語を抽出する
2. その共起の度合を表す尺度として対数尤度比を利用する

対数尤度比を利用した対訳候補の抽出法

	効率	効率以外	
efficiency	a	b	a+b
efficiency 以外	c	d	c + d
	a+c	b+d	n

a = 「効率」と「efficiency」が共に存在する対訳文の数
b = 「効率」がなく「efficiency」がある対訳文数
c = 「効率」があり「efficiency」がない対訳文数
d = 「効率」も「efficiency」も存在しない対訳文数
n = 全対訳文数

読売新聞と The Daily Yomiuri のデータでは

$$a = 137,\ b = 59,\ c = 284,\ d = 149520$$

もし,「効率」の存在が「efficiency」の存在に影響を与えないならば

$$\frac{a}{a+c} \sim \frac{b}{b+d} \tag{1}$$

のはずである．しかし，もし,「効率」と「efficiency」が良く共起するなら(あるいはあまり共起しないなら)

$$\frac{a}{a+c} \neq \frac{b}{b+d} \tag{2}$$

のはずである．
1 式は両者が確率的に独立であり，2 式は両者が確率的に従属であることを示す．→ 独立性の検定を利用し，独立性が低いものを抽出する．

対数尤度比検定 (Log-Likelihood Ratio Test)

$$LLR = \log \frac{P(\text{データ}\,|\,\text{従属})}{P(\text{データ}\,|\,\text{独立})} \tag{3}$$

を計算する．$LLR \gg 0$ならば,「効率」と「efficiency」については，従属であると考えた方が良いので，この値が大きければ，対訳候補として有望と考える．つまり各単語と「efficiency」について，LLR を計算し，その LLR が大きい単語を対訳候補とする．

対数尤度比の例

単語	対訳	a	b	c	d	LLR
efficiency	効率	137	59	284	149520	710
efficiency	化	79	117	6984	142820	115
efficiency	燃費	14	182	15	149789	73
efficiency	性	51	145	4861	144943	67
efficiency	向上	22	174	455	149349	58
decline	減少	139	487	525	148849	439
decline	低下	91	535	550	148824	245
decline	下落	81	545	414	148960	230
decline	減	56	570	245	149129	165
decline	連続	57	569	514	148860	131
commercial	商業	120	325	101	149454	564
commercial	銀行	131	314	1900	147655	302
commercial	都市	50	395	710	148845	111
commercial	捕鯨	26	419	83	149472	92
commercial	都銀	23	422	44	149511	91

データの表現法

P(データ | 独立) や P(データ | 従属) を計算するには，データを数値表現しないといけない．このときのデータの単位は対訳文である．そこで

$$E_i = \begin{cases} 1 \text{「efficiency」が対訳文} i \text{に出現する} \\ 0 \text{ 出現しない} \end{cases}$$

$$K_i = \begin{cases} 1 \text{「効率」が対訳文} i \text{に出現する} \\ 0 \text{ 出現しない} \end{cases}$$

という変数を定義する．すると

$$a = \sum_{i=1}^{n} [E_i = 1][K_i = 1] \tag{4}$$

$$b = \sum_{i=1}^{n} [E_i = 1][K_i = 0] \tag{5}$$

$$c = \sum_{i=1}^{n} [E_i = 0][K_i = 1] \tag{6}$$

$$d = \sum_{i=1}^{n} [E_i = 0][K_i = 0] \tag{7}$$

である．

$P(\text{データ}\,|\,\text{独立})$の計算

$$
\begin{aligned}
\log P(\text{データ}\,|\,\text{独立}) &= \sum_{i=1}^{n} \log P(E_i, K_i|\,\text{独立}) \\
&= a \log P(E_i = 1, K_i = 1|\,\text{独立}) \\
&+ b \log P(E_i = 1, K_i = 0|\,\text{独立}) \\
&+ c \log P(E_i = 0, K_i = 1|\,\text{独立}) \\
&+ d \log P(E_i = 0, K_i = 0|\,\text{独立})
\end{aligned}
$$

$$
\begin{aligned}
&\log P(E_i = 1, K_i = 1|\,\text{独立}) \\
&= \log P(E_i = 1|\,\text{独立}) P(K_i = 1|\,\text{独立}) \\
&= \log \frac{a+b}{n} \frac{a+c}{n}
\end{aligned}
$$

$$
\begin{aligned}
&\log P(E_i = 1, K_i = 0|\,\text{独立}) \\
&= \log P(E_i = 1|\,\text{独立}) P(K_i = 0|\,\text{独立}) \\
&= \log \frac{a+b}{n} \frac{b+d}{n}
\end{aligned}
$$

残りの2つについても同様

$P(\text{データ}\,|\,\text{従属})$の計算

$$
\begin{aligned}
\log P(\text{データ}\,|\,\text{従属}) &= \sum_{i=1}^{n} \log P(E_i, K_i|\,\text{従属}) \\
&= a \log P(E_i = 1, K_i = 1|\,\text{従属}) \\
&+ b \log P(E_i = 1, K_i = 0|\,\text{従属}) \\
&+ c \log P(E_i = 0, K_i = 1|\,\text{従属}) \\
&+ d \log P(E_i = 0, K_i = 0|\,\text{従属})
\end{aligned}
$$

$$\log P(E_i = 1, K_i = 1|\,\text{従属}) = \log \frac{a}{n}$$

$$\log P(E_i = 1, K_i = 0|\,\text{従属}) = \log \frac{b}{n}$$

残りの2つについても同様

問題 (15分)

$$LLR = \log \frac{P(\text{データ} \,|\, \text{従属})}{P(\text{データ} \,|\, \text{独立})}$$

を a, b, c, d, n により，なるべく簡単な形式で表現して下さい.

回答例

$$
\begin{aligned}
LLR &= a\log\frac{\frac{a}{n}}{\frac{a+b}{n}\frac{a+c}{n}} + b\log\frac{\frac{b}{n}}{\frac{a+b}{n}\frac{b+d}{n}} \\
&+ c\log\frac{\frac{c}{n}}{\frac{c+d}{n}\frac{a+c}{n}} + d\log\frac{\frac{d}{n}}{\frac{c+d}{n}\frac{b+d}{n}} \\
&= a\log a + b\log b + c\log c + d\log d \\
&+ (a+b+c+d)\log n \\
&- (a+b)\log(a+b) - (a+c)\log(a+c) \\
&- (b+d)\log(b+d) - (c+d)\log(c+d) \\
&= l(a) + l(b) + l(c) + l(d) + l(n) \\
&- l(a+b) - l(a+c) - l(b+d) - l(c+d)
\end{aligned}
$$

ただし，$l(x) = x\log(x)$

まとめ

- 対訳コーパスは，英語教育や日本語教育にも役立つ
- LLRを利用することにより，対訳候補を抽出できる

12. 対訳コーパスの自動作成

内山将夫@NICT
mutiyama@nict.go.jp

対訳コーパスの自動作成

対訳関係にある文章から
対訳関係にある文対応を同定することにより
対訳コーパスを自動作成する

対訳文対応同定の手順

1. 英語文章 E と日本語文章 J を用意する
2. E を辞書引きして，日本語単語に変換する
3. J を単語に分割する
4. 単語同士の対応を利用して，文同士の対応をみつける

日本語テキストの例

j0 天文学者

j1 ある天文学者は、夜になるとしょっちゅう、星を観測しに郊外へと出かけて行った。

j2 ある晩、彼は、星に気を取られていて、誤って深い井戸に落ちてしまった。

j3 彼は、打ち身や切り傷をつくって、悲鳴を上げた。

j4 その声を聞きつけて、近所の人が井戸へと飛んできた。

j5 そして何が起きたのかを知ると、こんな事を言った。

j6 「天国を覗くことばかりに、うつつを抜かしてないで、少しは足下に注意を払いなさいな」

英語テキストの例

e0 The Astronomer

e1 AN ASTRONOMER used to go out at night to observe the stars.

e2 One evening, as he wandered through the suburbs with his whole attention fixed on the sky,

e3 he fell accidentally into a deep well.

e4 While he lamented and bewailed his sores and bruises,

e5 and cried loudly for help,

e6 a neighbor ran to the well,

e7 and learning what had happened said:

e8 "Hark ye, old fellow, why, in striving to pry into what is in heaven,

e9 do you not manage to see what is on earth?

e10 "

日本語テキストを単語に分割し内容語をとる

```
j0 天文学 天文学
j0 者 者
...
j1 出かけ 出かける
j1 行っ 行く
j2 晩 晩
j2 彼 彼
...
j2 井戸 井戸
j2 落ち 落ちる
j3 彼 彼
...
j3 上げ 上げる
j4 声 声
j4 聞きつけ 聞きつける
j4 近所 近所
...
j5 起き 起きる
j5 知る 知る
j5 事 事
...
j6 注意 注意
j6 払い 払う
```

英語テキストを辞書引きする

e0 astronomer 者 天文学 astronomer
e1 go_out_at_night 戸出 夜 夜歩き
e1 observe マーク 意見 監視 観ずる 観る ...
e1 stars 星屑 星辰 天涯孤独 到達 不可能 ...
....
e2 fixed あてがう こわばる 一定 確固たる...
e2 sky お天気 たる ひばり スカイ ...
e3 fell すさまじい たくましい ぴりっと ...
...
e4 lamented 哀悼 故人 死者 惜しむ ...
e4 bewailed bewail 哀 号 号泣 愁傷 ...
e4 sores sore しゃくにさわる ただれる ...
...
e6 neighbor 近く 近所 近付ける 近傍 ....
e6 ran くつ下 ぶつける バス 引く ...
e6 well い いい す たんまり ...
e7 learning 憶 憶える 科学 会釈 ...
...
e8 striving 合 辛苦 戦 張 張りあい ...
e8 pry こじあける せんさく てこ ...
...
e9 see お目もじ ご覧 しばしば ...
...

文同士の対応をみつける

天文学者
The Astronomer
ある天文学者は、夜になるとしょっちゅう、星を観測しに郊外へと出かけて行った。
AN ASTRONOMER used to go out at night to observe the stars.
ある晩、彼は、星に気を取られていて、誤って深い井戸に落ちてしまった。
One evening, as he wandered through the suburbs with his whole attention fixed on the sky,
he fell accidentally into a deep well.
彼は、打ち身や切り傷をつくって、悲鳴を上げた。
While he lamented and bewailed his sores and bruises,
その声を聞きつけて、近所の人が井戸へと飛んできた。
and cried loudly for help,
a neighbor ran to the well,
そして何が起きたのかを知ると、こんな事を言った。
and learning what had happened said:
「天国を覗くことばかりに、うつつを抜かしてないで、少しは足下に注意を払いなさいな」
”Hark ye, old fellow, why, in striving to pry into what is in heaven,
do you not manage to see what is on earth?
”

どういう文同士の対応をみつけるか

可能な対応例はたくさんある
そのうちで最適なものを見つけたい

j0	e0
j1	e1
j2	e2, e3
j3	e4
j4	e5,e6
j5	e7
j6	e8,e9,e10

j0	e0,e1
j1,j2	e2, e3
j3	e4
j4	e5,e6,e7
j5,j6	e8,e9,e10

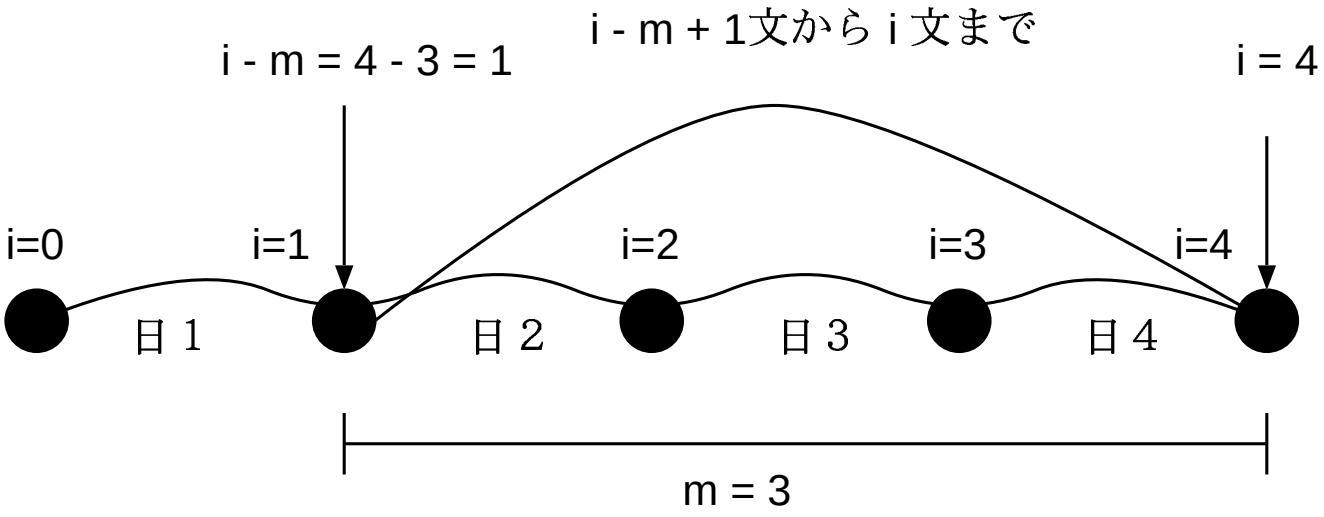
最適性の定義

$$\arg \max_{\text{文対応例} \in \text{可能な文対応集合}} \sum_{\text{各文対応}} \text{SIM(各文対応)}$$

- いくつもの文対応例があるが，そのうち，上式を満すものをとる
- このとき，SIMは，ある文対応例における，各文対応の類似度である.
- よって，上式における和は，ある文対応例のスコアとして,
- 各文対応の類似度の和を採用している.

要するに，類似度最大となるような文対応例を求める

数式表現



まず，複数文を表現するために，iが日文$_i$と日文$_{i+1}$の間にあるとすると

$$\text{日}(i-m,i) = \text{日}_{i-m+1}\text{日}_{i-m+2}\ldots\text{日}_{i}$$

$$\text{英}(j-n,j) = \text{英}_{j-n+1}\text{英}_{j-n+2}\ldots\text{英}_{j}$$

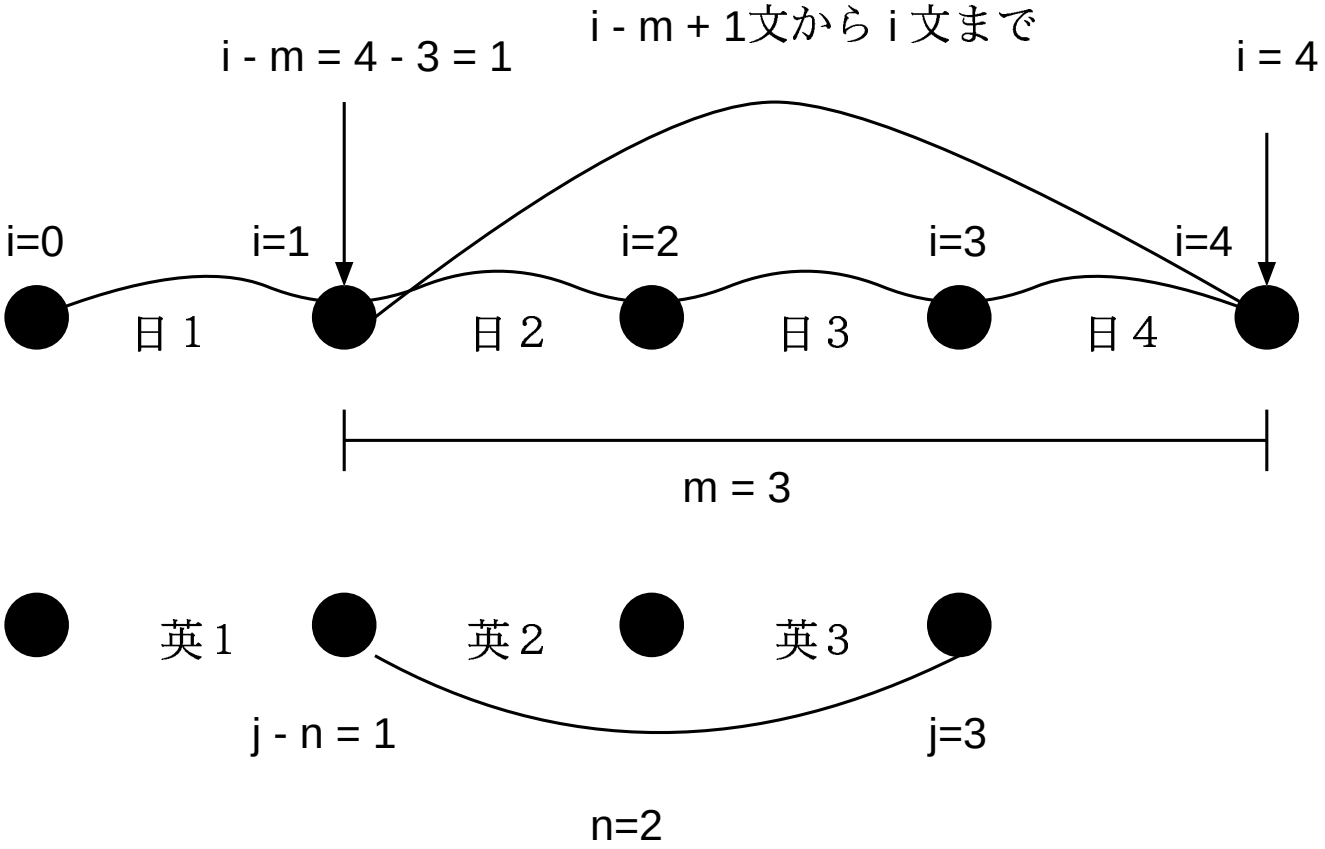
は，日$_i$以前のm単語と，英$_j$以前のn単語である．そうするとJを日本語文数，Eを日本語文数として，

$$\max \sum_{i=0}^{J} \sum_{j=0}^{E} \sum_{m=0}^{i} \sum_{n=0}^{j} \mathrm{SIM}(\text{日}(i-m,i), \text{英}(j-n,j))$$

となるような類似度の中で，ちゃんと対応の整合性がとれているようなものをとる．

再帰式による最大スコアの計算

今計算したいものは，$S(i,j)$である．
$S(i,j)$とは，日本語文がi文，英語文がj文並べられたときの最適スコアである．



このとき，$S(i-m,j-n)$が既に求まっていると仮定する．つまり，日本語の$i-m$文までと，英語の$j-n$文までは，対応済みとする．上例だと,「日$_1$日$_2$日$_3$日$_4$」と「英$_1$英$_2$英$_3$」のスコアを求めるとき，$m=3,n=2$のときには，日$_1$と英$_1$のスコアは計算済とする．すると

$$S(i,j) = S(i-m,j-n) + \mathrm{SIM}(\text{日}(i-m,i) + \text{英}(j-n,j))$$

により$S(i,j)$が求まる．

つまり

$$\begin{aligned} & S(\text{日}_1\text{日}_2\text{日}_3\text{日}_4,\ \text{英}_1\text{英}_2\text{英}_3) \\ & = S(\text{日}_1,\ \text{英}_1) + \mathrm{SIM}(\text{日}_2\text{日}_3\text{日}_4, \text{英}_2\text{英}_3) \qquad (1) \end{aligned}$$

である．これは，m と n が与えられている場合であるが，m と n については，固定されていないので，全ての m と n を考える必要がある．ただし，あまり多くすると計算量が多くなるので，たとえば，

$$(n, m) \in \{(1,0), (1,1), (1,2), (1,3), (0,1), (2,1), (3,1), (2,2)\}$$

とする．そのとき，

$$S(i,j) = \max_{m,n} S(i-m, j-n) + \mathrm{SIM}(\text{日}(i-m, i), \text{英}(j-n, j))$$

である．また，

$$S(0,0) = 0$$

なので，帰納的に最適スコアを定義できる．

$S(i,j)$の計算法

テーブルSを用意して，それを小さい方から埋めていく

$$日_1(a,b), 日_2(c), 日_3(d)$$

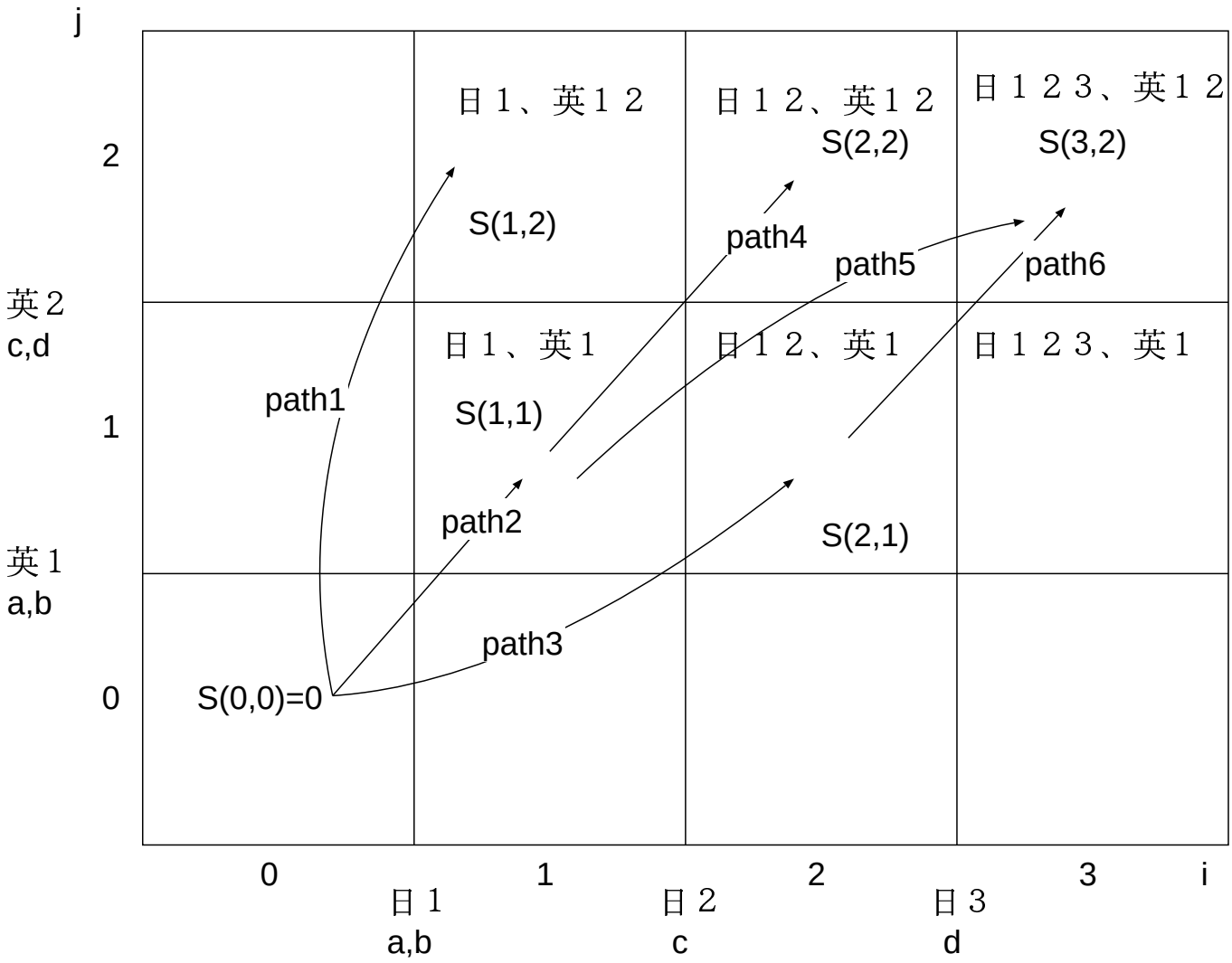
$$英_1(a,b), 英_2(c,d)$$

について，1対1，1対2，2対1の対応を許すときに，どのような文対応があるかをみる．a,b,c,dは単語とする．類似度は

$$\mathrm{SIM}(J,E) = \frac{2|J \cap E|}{|J| + |E|}$$

により計算する．ただし，J,Eは日英における単語集合．

テーブル S


j
2
1
0
英２
c,d
英１
a,b
日１、英１２
日１２、英１２
日１２３、英１２
S(2,2)
S(3,2)
S(1,2)
path4
path5
path6
日１、英１
日１２、英１
日１２３、英１
path1
S(1,1)
path2
S(2,1)
path3
S(0,0)=0
0
1
2
3
i
日１
a,b
日２
c
日３
d

テーブル S の埋め方

- まず，$S(0,0)=0$ とする

次に，$i,\ j$ の小さい方から $S(i,j)$ を埋めていく．このとき，1対1，1対2，2対1の文対応のみを許すので，図のパスのように，斜め方向に進む対応のみが許される．
したがって，
$i=1$ のときには，

path1 $\mathrm{SIM}(\text{日}\,1,\text{英}\,12) = \mathrm{SIM}(\{a,b\},\{a,b,c,d\}) = \frac{2\times2}{2+4} = \frac{4}{6} = 0.67$

path2 $\mathrm{SIM}(\text{日}\,1,\text{英}\,1) = \mathrm{SIM}(\{a,b\},\{a,b\}) = \frac{2\times2}{2+2} = 1$

より

$$
\begin{aligned}
S(1,1) &= S(0,0) + \mathrm{SIM}(\text{日}\,1,\text{英}\,1) = 1 \\
S(1,2) &= S(0,0) + \mathrm{SIM}(\text{日}\,1,\text{英}\,12) = 0.67
\end{aligned}
$$

$i = 2$のときには，

path3 $\mathrm{SIM}(\text{日}12, \text{英}1) = \mathrm{SIM}(\{a, b, c\}, \{a, b\}) = \frac{2\times2}{3+2} = \frac{4}{5} = 0.8$

path4 $\mathrm{SIM}(\text{日}2, \text{英}2) = \mathrm{SIM}(\{c\}, \{c, d\}) = \frac{2\times1}{1+2} = \frac{2}{3} = 0.67$

path1 から S(2,2) に移るためには，真横に移動しないといけないが，これは，日本語文は消費されるが，英語文は消費されないので，1対0 の対応となる．この対応は許されていない．よって，

$$S(2,1) = S(0,0) + \mathrm{SIM}(\text{日}12, \text{英}1) = 0.8$$
$$S(2,2) = S(1,1) + \mathrm{SIM}(\text{日}2, \text{英}2) = 1 + 0.67 = 1.67$$

$i = 3$のときには，

path5 $\mathrm{SIM}(\text{日}23, \text{英}2) = \mathrm{SIM}(\{c, d\}, \{c, d\}) = \frac{2\times2}{2+2} = 1$

path6 $\mathrm{SIM}(\text{日}3, \text{英}2) = \mathrm{SIM}(\{d\}, \{c, d\}) = \frac{2\times1}{1+2} = \frac{2}{3} = 0.67$

よって，path5 を通ったときには

$$S(3, 2) = S(1, 1) + \mathrm{SIM}(\text{日}23, \text{英}2) = 1 + 1 = 2$$

path6 を通ったときには

$$S(3, 2) = S(2, 1) + \mathrm{SIM}(\text{日}3, \text{英}2) = 0.8 + 0.67 = 1.47$$

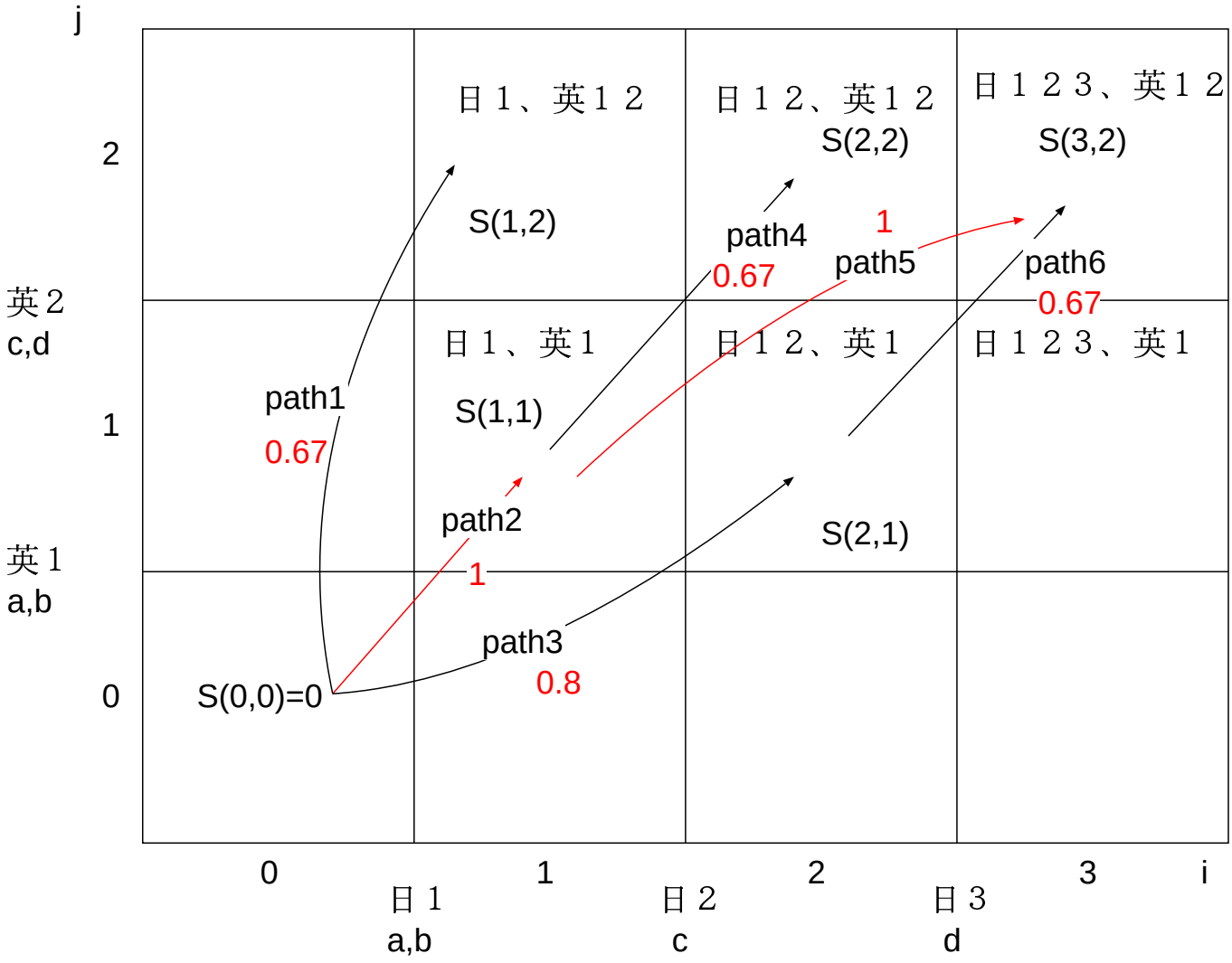
これらの最大値をとって，

$$S(3, 2) = 2$$

$S(0,0)$から path2 と path5 を通って $S(3,2)$ に至るパスが最適解である．そのときの対応は，

$$\begin{aligned} S(3,2) &= S(1,1) + \mathrm{SIM}(\text{日}23, \text{英}2) \\ &= S(0,0) + \mathrm{SIM}(\text{日}1, \text{英}1) + \mathrm{SIM}(\text{日}23, \text{英}2) \\ &= 0 + 1 + 1 = 2 \end{aligned}$$

より，「日1と英1」「日23と英2」が対応している．



文対応のまとめ

日本語文数をJ，英語文数をEとすると

$$S(J, E)$$

は最大スコアによる文対応のスコアとなる
このとき，各文対応を$a_1, a_2, ..., a_N$とすると

$$S(J, E) = \sum_{i=1}^{N} \mathrm{SIM}(a_i)$$

ただし，a_iは，前述の path にあたると考えてよく，$\mathrm{SIM}(a_i)$は，対応a_iを構成する日本語文と英語文の類似度である.
このとき，

$$\mathrm{AVSIM} = \frac{S(J, E)}{N}$$

は，各対応a_iの類似度SIMの平均値である.

対訳コーパスの自動構築

新聞記事のように，必ずしも直訳されていない対訳テキスト$T_1, T_2, ...$があるとする．
各 T_i について，文対応を求めると，それにより$\mathrm{AVSIM}(T_i)$が求まる．
$\mathrm{AVSIM}(T_i)$が大きいT_iは良く似た文対応からなると考えられる．
したがって，この値の大きいテキストをとることにより，対訳の度合が高いテキストを取ることができる．
また，良い対応のテキストに含まれる文は，良い文対応だと考えられるので，

$$\text{文対応のスコア} = \mathrm{SIM} \times \mathrm{AVSIM}$$

は，テキストの類似度までを考慮した文対応スコアである．
この文対応スコアが大きいものから対訳コーパスに採用する．

対訳コーパスについてのまとめ

- 現状で利用可能な対訳コーパスとして，NICTで公開しているものを紹介した．
- 対訳コーパスは機械翻訳以外にも使えることを示した．
- 対訳テキストから対訳コーパスを自動作成する方法を示した．

翻訳モデルと翻訳エンジン

内山将夫@NICT
mutiyama@nict.go.jp

これまでのまとめ

- MT の性能評価
- 言語モデル
- 単語対応のモデル化
- 対訳コーパスの作成

これからの目次

- 翻訳モデルと翻訳エンジンの関係
- 句(フレーズ)に基づく翻訳モデル
- 句表(フレーズテーブル)の作成
- 対数線形モデルとデコーダー
- 誤り率最小化訓練 (MERT, minimum error rate training)

翻訳モデルと翻訳エンジンの関係

- 翻訳エンジンというのは，入力文$\mathbf{f}$を受け取り，それを翻訳して，出力文$\hat{\mathbf{e}}$とするものである．
- そのときには，何らかの探索アルゴリズムを利用して，最適スコアの出力文$\hat{\mathbf{e}}$を得る．
- そのスコアの付け方gを翻訳モデルという

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} g(\mathbf{e}, \mathbf{f})$$

  - $\hat{\mathbf{e}}$は出力文
  - $\arg\max_{\mathbf{e}}$は最適解の探索を示す
  - $g(\mathbf{e}, \mathbf{f})$が翻訳モデル
- 翻訳エンジンは，翻訳モデルに従って，最適な翻訳を得るためのものである．

探索アルゴリズムと翻訳モデルの密接な関係

1. 出力候補の数はとても多い.

入力 $\mathbf{f} = f_1 f_2 ... f_m$ が，それぞれ，3単語の対訳単語をもつとすると，それだけで，3^m の組み合わせがある．さらに順番が入れかわるので，$m!3^m$ の組み合わせがある．これは，単語が1対1に翻訳されるとした場合でもである．

m	1	3	5	7	10
$m!3^m$	3	162	29160	1千万	2000億

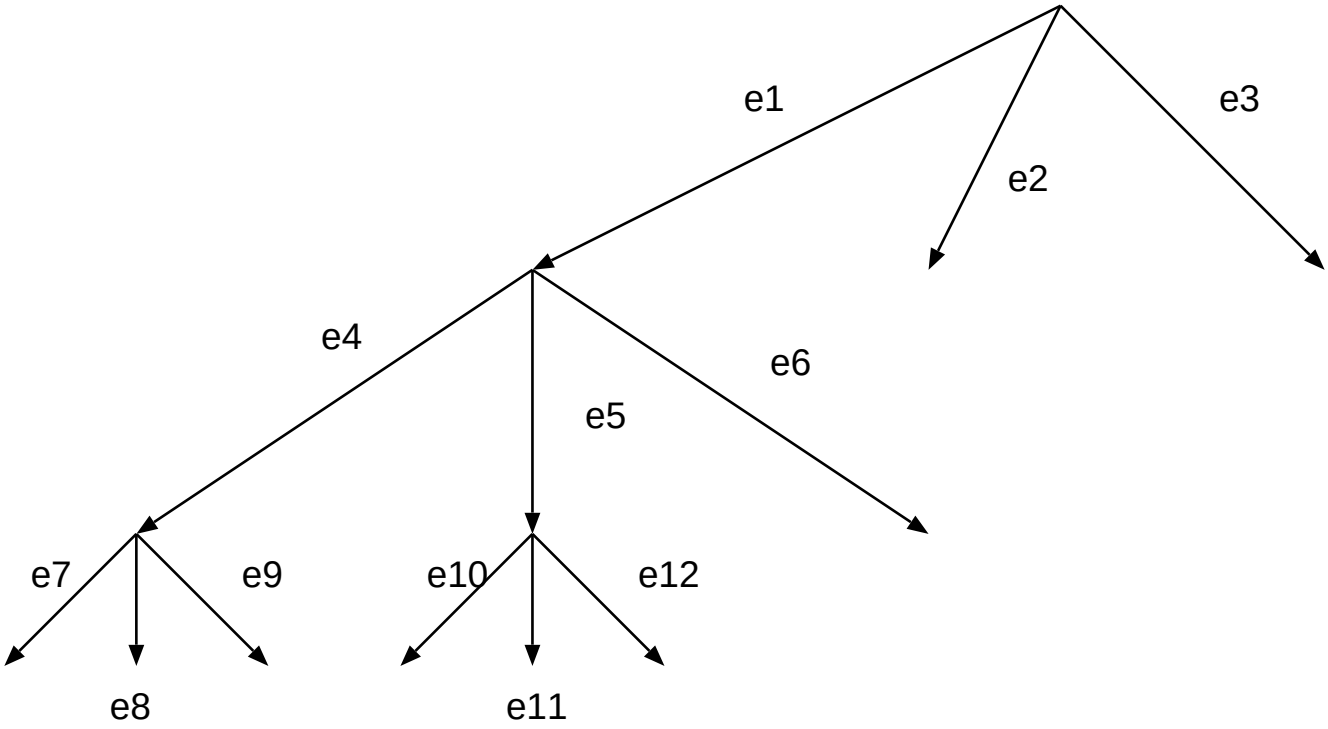
のように単語数mに応じて，すぐに大きくなる．

→ 全部を列挙することは不可能

探索アルゴリズムと翻訳モデルの密接な関係

2. 探索は翻訳モデルにガイド される

少しずつ出力文をつくっていき，その都度，翻訳モデルにより割当てられるスコアの小さい候補を消していく



良い翻訳モデルならば良いスコアを与えることができるので，いらない候補を早く消すことができる．また，スコアの計算方法と探索法とには，密接な関係がある．
→ 実際上スコアの計算法は，探索方法により決まる

ただし，たとえば，混合整数計画法により，スコアを定義することも可能なので，そのうちに，探索法は，MTの研究の枠外になるかもしれない．

統計的翻訳モデルの歴史概観

1. 単語単位の翻訳モデル (1980年代後半〜1990年代)
2. 句単位の翻訳モデル (1990年代後半〜)
3. 構造に基づく翻訳モデル (2000年代半ば〜)

2や3の始まりは，1に少し遅れた時期だが，研究が盛んになったのは，それぞれ，2000年〜，2005年〜くらいである．現在は，構造に基づく翻訳モデルに研究の主流は移ってきている．
しかし，句単位の翻訳は，簡単でかつ効率的で，構造に基づく翻訳の基礎となるものなので，ここでは，句単位の翻訳モデルを紹介する．

14. 句単位の翻訳モデルの概要

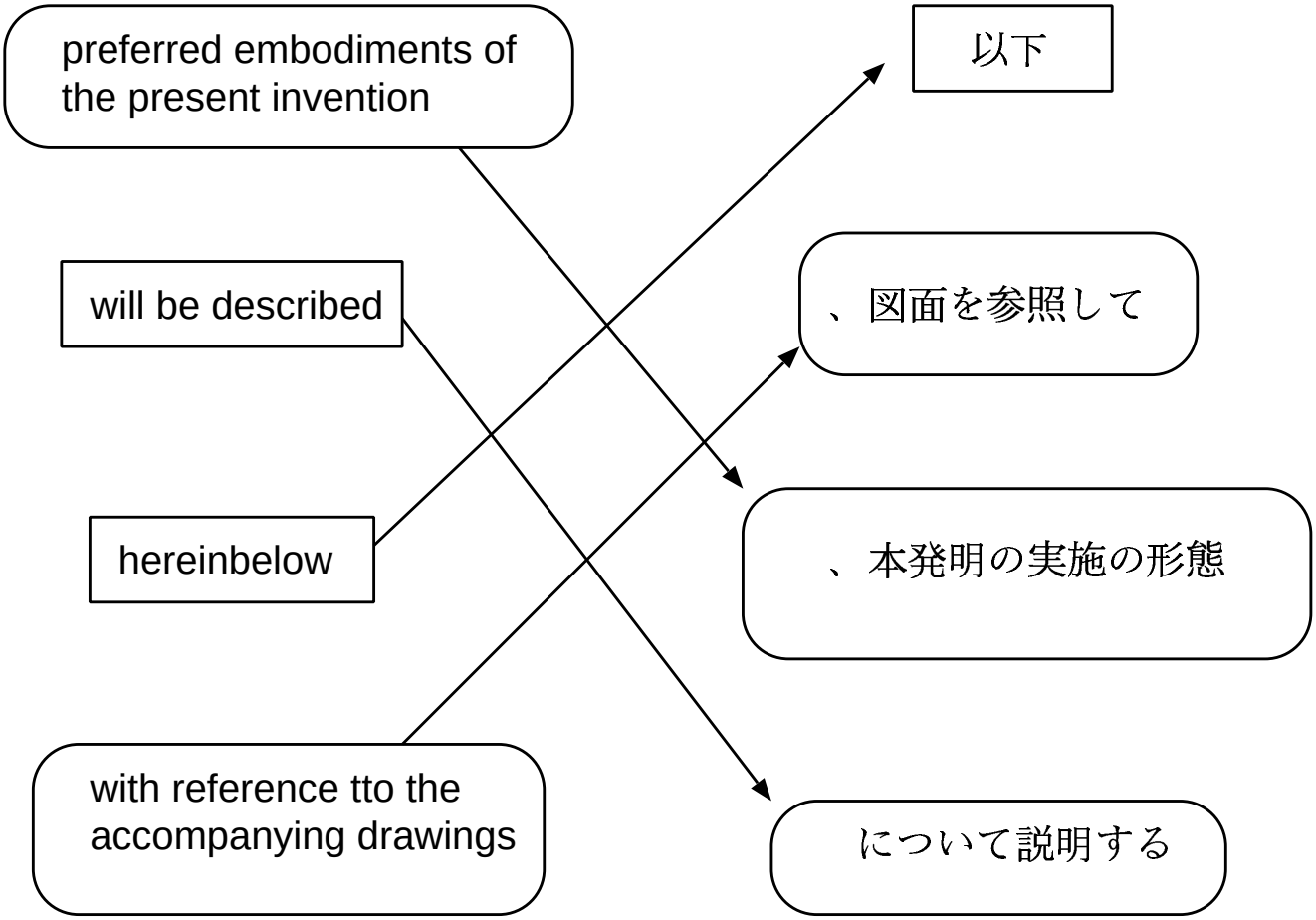
内山将夫@NICT
mutiyama@nict.go.jp

句単位の翻訳 (Phrase-based SMT)

- ここでいう句(フレーズ)とは，任意の連続する単語列のことであり，名詞句や動詞句などの言語学的な単位とは無関係である.
- 句単位の翻訳は
  - 入力文を句に分割する
  - 各句を翻訳する
  - 翻訳した句を並べかえる

  ことによりできる

句単位の英日翻訳の例



- 英語の文を句にわける
  (e.g., will be described)
- 各句を翻訳する
  (e.g., について説明する)
- 句を並べ替える

句単位の翻訳に必要なもの

1. 句の翻訳表 (フレーズテーブル)
   英句1 和句1
   英句2 和句2
   …
2. 翻訳候補へのスコア付けの方法

まず，1について説明する．

15. フレーズテーブルの作り方

内山将夫@NICT
mutiyama@nict.go.jp

フレーズテーブルの作成手順

1. 初期単語アラインメント
2. 単語アラインメントの修正
3. 単語単位の翻訳確率の推定
4. フレーズ対応の抽出
5. フレーズへのスコアの付与

結果

英語句 ||| 日本語句 ||| スコア1 スコア2 ...

これらの句のスコアを上手く組み合わせて文全体のスコアを得る.

初期単語アラインメント

GIZA++ というフリーソフトウェアを利用することが多い．これを，たとえば，日英の方向に，

- IBM Model-1 (紹介済)
- HMM Model
- IBM Model-3
- IBM Model-4

を順番に適用していくことにより，日英の各文対応について，1対nの単語対応を得ることができる．
英日にもすることにより，両方向に，1対nの単語対応を得る．

==========

単語アラインメントは未解決の問題だが，ある程度まではできるので，とりあえず，そこまでを前提にして，次に進んでいる．

日英単語アラインメントの例

	when	the	fluid	pressure	cylinder	31	is	used	,	fluid	is	gradually	applied	.
流体			*											
圧				*										
シリンダ					*									
３１						*								
の														
場合	*													
は							*							
流体										*				
が											*			
徐々に												*		
排出												*		
さ													*	
れる													*	
こと													*	
と									*					
なる														
。														*

日英単語アラインメントの例

	this	relation	must	be	maintained	even	after	passing	at	least	100,000	sheets	.
そして							*						
、				*									
上記	*												
関係		*											
を								*					
少なくとも										*			
１０万											*		
枚											*		
通											*		
紙												*	
し								*					
て						*							
も						*							
維持					*								
し								*					
なけれ			*										
ば			*										
なら			*										
ない			*										
。													*

日英単語アラインメントの例

	first	,	a	description	will	be	given	of	the	structure	of	the	wheels	2	.
まず	*														
、		*													
車輪													*		
2														*	
の											*				
構造										*					
について					*										
説明					*										
する					*										
。															*

英日単語アラインメントの例

	流体	圧	シリンダ	３１	の	場合	は	流体	が	徐々に	排出	さ	れる	こと	と	なる
when						*										
the							*									
fluid	*															
pressure		*														
cylinder			*													
31				*												
is													*			
used															*	
,																
fluid								*								
is									*							
gradually										*						
applied											*					
.																

英日単語アラインメントの例

	まず	、	車輪	2	の	構造	について	説明	する	。
first	*									
,		*								
a										
description								*		
will								*		
be								*		
given								*		
of					*					
the					*					
structure						*				
of							*			
the									*	
wheels			*							
2				*						
.										*

日英英日アラインメントの合成

- アラインメントの積集合 = I
  → 精度の高い単語対応
- アラインメントの積集合 = U
  → カバー率の高い単語対応

I を出発点として U 中のアラインメントを加えていく.

日英と英日の合成の例

	when	the	fluid	pressure	cylinder	31	is	used	,	fluid	is	gradually	applied	.
流体			I											
圧				I										
シリンダ					I									
３１						I								
の														
場合	I													
は		U					U							
流体										I				
が											I			
徐々に												I		
排出												U	U	
さ													U	
れる							U						U	
こと													U	
と								U	U					
なる														
。														I

日英と英日の合成の例

	this	relation	must	be	maintained	even	after	passing	at	least	100,000	sheets	.
そして							U						
、				U									
上記	I												
関係		I											
を								U					
少なくとも									U	I			
１０万											I		
枚											U	U	
通								U			U		
紙												U	
し							U	U					
て						U							
も						I							
維持					I								
し								U					
なけれ			U										
ば			U										
なら			I	U									
ない			U										
。													I

日英と英日の合成の例

	first	,	a	description	will	be	given	of	the	structure	of	the	wheels	2	.
まず	I														
、		I													
車輪													I		
2														I	
の								U	U		U				
構造										I					
について					U						U				
説明				U	I	U	U								
する					U							U			
。															I

合成手順

```
#
# アラインメント候補 = {(e_i, f_j)} は e_i + fj が小さいものから並
# んでいる．そうすることにより，文頭のものから優先して対応付けてい
# くことができる．
#

日英英日アラインメントの合成 ()
  隣接点集合 = ((-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1))
  アラインメント候補 = 積集合 I(日英アラインメント，英日アラインメント)
  隣接する 1 対 1 の点を追加する ()
  その他の点を加える ()
  アラインメント候補を返す
end

隣接する 1 対 1 の点 N を追加する ()
  while true
    for (j, e) ∈ アラインメント候補
      for (j0, e0) ∈ 隣接点 (j,e) ∩ 和集合 (日英，英日)
         if(j0 も e0 もアラインメント候補で使われていない)
           (j0, e0) をアラインメント候補に追加する (そうしても 1 対 1 が壊れない)
         end
      end
    end
    アラインメント候補に追加がなければ終了する
  end
end

#
# なるべく多くの点を加えたいが，日英共に使われているのは加えない
#
その他の点 O を加える ()
  for (j,e) ∈ 和集合 (日英，英日)
    if (j か e のどちらかがアラインメント候補で使われていない)
      (j,e) をアラインメント候補に含める
    end
  end
end
```

合成の例

(I=積集合，N=隣接点，O=その他の追加, U=削除された点)

	when	the	fluid	pressure	cylinder	31	is	used	,	fluid	is	gradually	applied	.
流体			I											
圧				I										
シリンダ					I									
31						I								
の														
場合	I													
は		N					O							
流体										I				
が											I			
徐々に												I		
排出												U	N	
さ													O	
れる							U						O	
こと													O	
と								O	O					
なる														
。														I

合成の例

	this	relation	must	be	maintained	even	after	passing	at	least	100,000	sheets	.
そして							O						
、				O									
上記	I												
関係		I											
を								O					
少なくとも									O	I			
１０万											I		
枚											U	N	
通								U			O		
紙												O	
し							U	O					
て						O							
も						I							
維持					I								
し								O					
なけれ			O										
ば			O										
なら			I	U									
ない			O										
。													I

合成の例

	first	,	a	description	will	be	given	of	the	structure	of	the	wheels	2	.
まず	I														
、		I													
車輪													I		
2														I	
の								O	N		U				
構造										I					
について					U						N				
説明				O	I	O	O								
する					O							O			
。															I

単語単位の翻訳確率

- これまでに，日英，英日の1対nおよびm対1の単語対応を組み合わせてm対nの単語対応を得る方法を示した.
- これから，それを利用して，フレーズを抽出する方法を示す.
- その前に，ここで得たm対nの対応から単語単位の辞書を作る方法を示す.

単語単位の辞書

アライメント

	e1	e2	e3	e4
j1	*		*	
j2				
j3		*		
j4				*

から，(j1,e1), (j1,e3), (j3, e2), (j4,e4) の単語対応がとれるので，これから，j $\rightarrow$ e と e $\rightarrow$ j 方向の辞書がつくれる．更に，

$$P(j|e) = \frac{\#(j,e)}{\Sigma_{j'}\#(j',e)}$$

$$P(e|j) = \frac{\#(j,e)}{\Sigma_{e'}\#(j,e')}$$

により，単語単位の翻訳確率が求まる．(スムージングをしても良い)

単語単位の辞書：頻度 日本語単語 英語単語

2230457 、 the
1811473 、 ,
1804260 。 .
1802874 の of
1767572 の the
704450 に to
668066 は the
584253 が is
543500 は is
527825 に in
480009 する the
448118 を is
376711 と and
376195 、 a
365107 図 fig.
345039 を of
287703 ） ）
285186 （ （
281659 信号 signal
272811 を to
245721 データ data
230038 回路 circuit
224630 で in
209459 で ,
202160 に on
198721 １ 1
196317 から from
193708 この the
177043 に shown
175086 し the
166888 、 and
164284 た the
154881 は a
146977 さ to
146509 ２ 2
143482 を for
142192 し to
140494 出力 output
139684 制御 control
135600 が are
130609 は are
128437 この this
126068 実施 embodiment
124007 ３ 3
117825 する to
112598 よう as
111611 た a
110575 ステップ step
109286 ４ 4
106286 および and
105113 を are
101722 で by
101291 及び and
100440 示す shown
98899 が of
98684 が can
97290 示す in
96570 ５ 5
95364 １０ 10
95330 電圧 voltage
93903 する be
93619 入力 input
92337 さ in
89141 形成 formed
87623 する a
87410 において in
85936 １ first
85471 その the
84919 として as
84888 第 first
84615 に with
84356 メモリ memory
82459 ６ 6
82098 値 value
81423 に ,
79752 情
報 information
79287 １２ 12
79126 １１ 11
79048 の for
78406 例 embodiment
78238 に at
78233 こと can
77678 接続 connected
77671 、 an
77513 部 portion
75575 画像 image
75000 ／ /
73687 に into
72920 上記 the
72590 層 layer
72466 第 second
72399 ７ 7
72231 膜 film

単語単位の辞書：$P(日|英)$でソート

emu emu
SCLK sclk1
+ .sub.+
ウェル -well
LRD lrd
LAT lat.sub.
MRSOUT mrsout
0 0 0 0 0 0 0 0 00000000
WDR wdr
CBS cbs.sub.
PxR pxr
MBT mbt
MBQ mbq
BRE bre
685 685
. 38q
ramarea ramarea
kDa kda
Sdw s.sub.dw
RHIT rhit
CS cs.sub.
dspls dspls
ZODL zodl
GCF gcf
7030 7030
pretrn pretrn
dqmx dqmx
2 2.sup.
wreq wreq
RWC rwc
SGP sgp
3080 3080
排気 exhaling
カテプシン cathepsin
WCBRB wcbrb
Vpk vpk
Vfv vfv
TRBL trbl
PDRHIT pdrhit
Cts c.sub.ts
4080 4080
試験 under-test
romarea romarea
RMIS rmis
PG pg.sub.
CTRLR ctrlr
311 3.sub.11
3090 3090
2041 204.sub.1
10070 10070
セルフプリチャージ self-precharge
waitid waitid
ZMR zmr
Vpass vpass
OPRH oprh
LIOi lioi
1916 1916
日 date-indicating
rblj rblj
ZCO zco
WSEL wsel.sub.
VTTAO vttao
RSRav rsr.sub.av
PDRTPi pdrtpi
Nsa nsa.sub.
Iab i.sub.ab
Cdx cd.sub.x
× .times..sigma..sub.i3
inode inode
SENPn senpn
SDBn sdbn
PFAi pfai
MOPLS mopls
K F L E A K M X kfleakmx
CAD cad7
980 980.degree.
8510 8510
831 83.sub.1
10 ta.sub.10
偏り one-sidedness
微動 micromotion
q se.sub.1-q
ment ment
e E C O N O M Y eeconomy
TRRQ trrq
RSDF rsdf
RPRG rprg
QHN qhn
IRQ irq9
HNMOS hnmos
GLj glj
EAV eav
DWDE dwde
Bjb bjb
BRIa bria
5022 5022
3122 3122

単語単位の辞書：$P(英|日)$でソート

オーディオ audio
ｐｃｈＭＯＳＦＥＴ pch-mosfet
ＩＩＤＲ iidr
@ @
ｃｏａｒｓｅ coarse
ｉｒｏｍ irom
Ｖｃｃｐ vccp
ＧＤＩ gdi
ＭＳＫｉ mski
ＣＫＭ ckm
ＮＰＲ npr
ＳＰＬ spl
ＲＨＩＴ rhit
ＮＯＸ .sigma.nox
ＺＭＣＨＧ zmchg
ＳＴＸ stx
ＲＩＴ rit
ＧＢＵＦＲ gbufr
ｅｘｔＺＲＡＳ extzras
ＭＮＰ mnp
コースター coaster
Ｓｕｐ s.sub.up
９１００ 9100
ｅｘｔＺＣＡＳ extzcas
ＳＨＬ shl
Ｉｎｚ in.sub.z
Ｉｍｓ i.sub.ms
ＢＬｒ blr
フォーマツタ formater
ｄｉｇｅｓｔ digest
ＲＸＴｐｕｌｓｅ rxtpulse
ＤＲａ dra
ＷＢＬｎ wbln
ＳＰＸ spx
ＳＥＬＦＯＳＣ selfosc
ＦＭＴ fmt
ＣＡＳＩ casi
ＰＲＥＤ pred
ＤＴＧ dtg
Ｃｏｐｙ copy
ＣＡａ caa
ｔＰＧＲ tpgr
ｄｑｍｘ dqmx
ＸＳＵ xsu
Ｖｒｃ vrc
ＶｃｏｍＨ vcomh
Ｓｌｕ slu
ＤＢＢａ dbba
ｍｐｇｍ mpgm
ｆＡ fa
ＸＢＬＡ xbla
ＲＥＭＥ reme
ＭＢＷ mbw
Ｃｌｔ clt
ＣＯＣ coc
ＣＬＥ cle
ＲＷＬＥ rwle
ＲＢＬｎ rbln
ＮＢＣ nbc
ＢＴＧ btg
ＢＧＭＭ bgmm
８２，８２ 82
スレッシュ threshold
ｑｃ qc
ｏａｍ oam
ｍｅｔａｐｈｏｒ metaphor
ＴＨｉｄｌ th.sub.idl
ＳＰＫＵ spku
ＲＭＢ rmb
ＲＦＩＣ rfic
ＥＸＴＣＬＫ extclk
ＢＬｈ blh
４０９０ 4090
ｎＭＩＳ nmis
ｌｕｎｃｈｅｒ luncher
ａｕｔｏｍａｔｏｎ automaton
Ｖｈｉｇｈ vhigh
ＳＬＩＦ slif
ＧＷＤ gwd
ＧＦＭ gfm
ＥＡＤ ead
ＤｉｎＬＢｎ dinlbn
ＣＷＬ cwl
ＣＴＲＬＲ ctrlr
ＢＳＢ bsb
１，５５５ 1,555
ｗａｉｔｉｄ waitid
ｖｂａｔ vbat
ｆｕｐ f.sub.up
ＸＳＤ xsd
ＶＺ vz
ＳＡＮＧ sang
ＭＩＳＦＥＴＱｈ q.sub.h

単語単位の辞書：対数尤度比でソート

。 .
の of
、 ,
、 the
の the
図 fig.
信号 signal
）)
（ (
データ data
に to
が is
回路 circuit
と and
は is
１ 1
から from
に in
出力 output
制御 control
２ 2
３ 3
を is
４ 4
電圧 voltage
実施 embodiment
１０ 10
ステップ step
５ 5
入力 input
メモリ memory
情報 information
１２ 12
６ 6
１１ 11
値 value
この this
は the
層 layer
発明 invention
７ 7
／ /
画像 image
膜 film
８ 8
方向 direction
２０ 20
１４ 14
に on
および and
する the
１３ 13
処理 processing
アドレス address
位置 position
、 a
形成 formed
及び and
で in
ゲート gate
９ 9
１６ 16
接続 connected
第 first
よう as
電流 current
電極 electrode
２１ 21
１５ 15
本 present
動作 operation
２２ 22
部 portion
基板 substrate
半導体 semiconductor
として as
図 figs.
３０ 30
示す shown
トランジスタ transistor
例 embodiment
に shown
第 second
レベル level
端子 terminal
装置 apparatus
装置 device
を of
状態 state
記録 recording
１ first
ない not

$P(*|\text{流体})$ と $P(*|\mathbf{fluid})$

```
# P(*|流体)
流体 fluid 0.84868421
流体 hydraulic 0.01946272
流体 gas 0.01151316
流体 fluids 0.01096491
流体 flow 0.01041667
流体 liquid 0.00849781
流体 liquid-pressure 0.00603070
流体 hydrodynamic 0.00575658
流体 working 0.00301535
流体 piston 0.00274123
```

```
# P(*|fluid)
fluid 流体 0.45072063
fluid 液 0.23875382
fluid 油 0.10642015
fluid 液体 0.02576794
fluid ブレーキフルード 0.01892561
fluid 流動 0.01368467
fluid 油圧 0.00873490
fluid 流 0.00844373
fluid 作動 0.00698792
fluid 連 0.00684234
```

フレーズ対応の抽出

これまでに

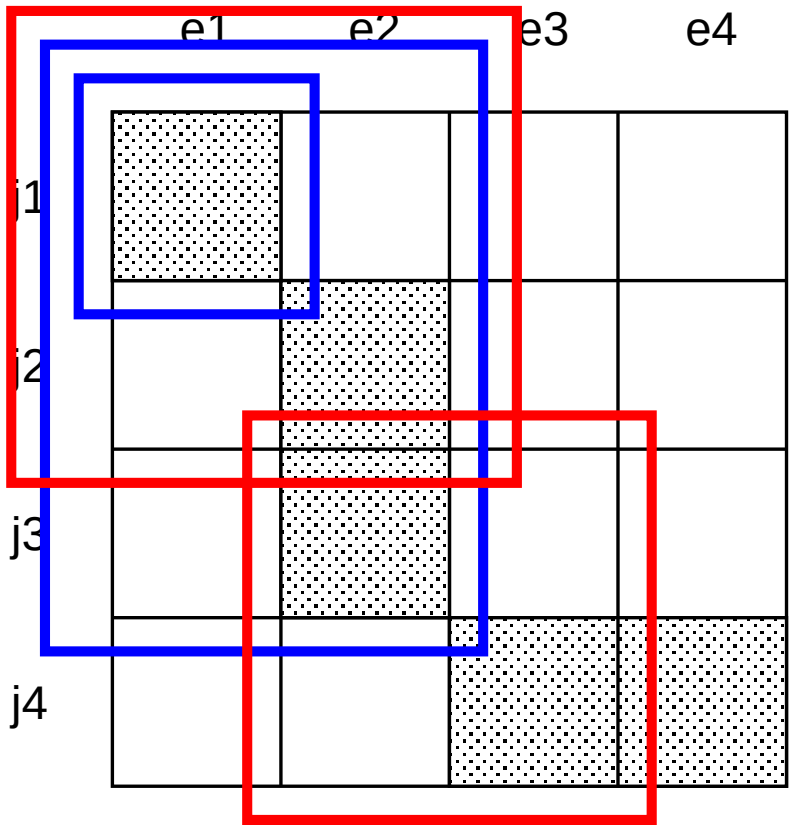
	e1	e2	e3	e4
j1	*			
j2		*		
j3		*		
j4			*	*

のような単語対応を得る方法を述べた．次は，ここから句対応を得る方法を述べる．

基本方針

単語対応に整合するような，なるべく多くの句対応を得る．

整合的な句の例



青はＯＫ．赤はダメ．
日本語の句Jと英語の句Eについて，$j \in J$の対応先を$e(j)$とすると，$e(j) \in E$でないといけない．
同様に$e \in E$の対応先を$j(e)$とすると，$j(e) \in J$である．
つまり，対応の相手は，日本語句Ｊ内の単語と英語句Ｅ内の単語に限られる．

句対応の例：fluid を含む句の例

(流体 ||| (fluid ||| 0.166667 0.0307503 1 0.623726 2.718
(流体 圧) ||| (fluid pressure) ||| 0.5 0.000742744 1 0.200361 2.718
(流体 圧 ||| (fluid pressure ||| 0.5 0.010301 1 0.266855 2.718
(流体 圧 源) ||| (fluid pressure source) ||| 1 0.000104349 1 0.159524 2.718
(流体 圧 源) １７ ||| (fluid pressure source) 17 ||| 1 9.465e-05 1 0.132599 2.718
(流体 圧 源 ||| (fluid pressure source ||| 1 0.00144721 1 0.212466 2.718
) の 送 液 |||) of fluid ||| 1 2.88816e-05 1 0.0150938 2.718
) の 送 液 が |||) of fluid has ||| 1 6.6955e-06 1 0.000451654 2.718
) の 送 液 が あっ た |||) of fluid has taken ||| 1 2.55555e-09 1 7.43761e-07 2.718
、 ３ は 液 圧制 御 装置 ||| , a fluid pressure control apparatus 3 ||| 1 2.28706e-07 1 0.000292281 2.7
、 ３ は 加工 液 ||| , 3 a dielectric fluid ||| 1 3.63074e-05 1 3.96712e-06 2.718
、 ３ は 加工 液 、 ||| , 3 a dielectric fluid , and ||| 1 1.47644e-05 1 5.67333e-08 2.718
、 ３８ は 加工 液 ||| , 38 is working fluid ||| 1 0.0088037 1 0.000662696 2.718
、 ３８ は 加工 液 供給 ||| , 38 is working fluid supplying ||| 1 0.0062226 1 4.34916e-05 2.718
、 ３８ は 加工 液 供給 手段 ||| , 38 is working fluid supplying means ||| 1 0.00427184 1 3.16053e-05 2
、 ４１ ｄ 、 流体 ||| , 41d , fluid ||| 1 0.0345864 1 0.000467132 2.718
、 ４１ ｄ 、 流体 貯留 ||| , 41d , fluid storing ||| 1 0.00048108 1 8.14764e-05 2.718
、 ４１ ｄ 、 流体 貯留 タンク ||| , 41d , fluid storing tank ||| 1 0.000284362 1 7.14906e-05 2.718
、 ６ および 流体 ||| and 6 and a fluid ||| 1 0.00410477 1 0.00964129 2.718
、 ６ および 流体 測 温 ||| and 6 and a fluid temperature measuring ||| 1 7.86306e-06 1 0.000721089 2.7
、 ６１ ’は 伝 熱 流体 ||| , 61 ’ a heat-transfer fluid ||| 1 0.00216443 1 0.000518098 2.718
、 ６１ は 伝 熱 流体 ||| , 61 is a heat-transfer fluid ||| 1 0.0185343 1 0.00119982 2.718
、 ７７ は 加工 液 ||| , 77 is a working fluid ||| 1 0.00864883 1 0.00028376 2.718
、 ７７ は 加工 液 ノズル ||| , 77 is a working fluid nozzle ||| 1 0.00764329 1 0.000210182 2.718
、 ７８ は 加工 液 ||| , and 78 is a working fluid ||| 1 0.00521192 1 1.09199e-05 2.718
、 Ｐ は 流体 ||| , p denotes fluid ||| 1 0.122825 1 0.000660778 2.718
、 Ｑ は 流体 の 流量 ||| , q denotes flow rate of fluid ||| 1 0.0145985 1 2.77106e-05 2.718
、 および 、 第 ２ 液 ||| and a second fluid ||| 0.333333 9.53798e-05 1 0.00106579 2.718
、 および 、 第 ２ 液 圧 ||| and a second fluid pressure ||| 0.5 3.19513e-05 1 0.00045599 2.718
、 この 液 ||| , and this fluid ||| 1 0.0557024 0.166667 0.000360561 2.718
、 この 液 ||| , the fluid ||| 0.0285714 0.0047175 0.166667 0.014136 2.718
、 この 液 ||| , this fluid ||| 1 0.0934985 0.333333 0.00959799 2.718
、 この 液 ||| 8 , and this fluid ||| 1 0.0557024 0.166667 8.11983e-08 2.718
、 この 液 圧 ||| , and this fluid pressure ||| 1 0.0186597 0.2 0.000154263 2.718
、 この 液 圧 ||| , the fluid pressure ||| 0.25 0.00158031 0.2 0.00604794 2.718
、 この 液 圧 ||| , this fluid pressure ||| 1 0.031321 0.4 0.00410641 2.718
、 この 液 圧 ||| 8 , and this fluid pressure ||| 1 0.0186597 0.2 3.47399e-08 2.718
、 この 液 圧 は ||| , and this fluid pressure is ||| 1 0.00536829 0.333333 4.16064e-05 2.718
、 この 液 圧 は ||| , the fluid pressure is ||| 0.5 0.000454647 0.333333 0.0016312 2.718
、 この 液 圧 は ||| 8 , and this fluid pressure is ||| 1 0.00536829 0.333333 9.36976e-09 2.718
、 この 液 圧 を ||| , this fluid pressure is ||| 1 0.00774038 1 0.000828222 2.718
、 この 作動 液 ||| , the hydraulic fluid ||| 0.125 0.000293068 1 0.000462624 2.718
、 この 磁性 流体 ||| , the magnetic fluid ||| 0.333333 0.00118769 0.5 0.0798598 2.718
、 この 磁性 流体 ||| the magnetic fluid ||| 0.0769231 0.000586521 0.5 0.239424 2.718
、 この 磁性 流体 ７３ ||| , the magnetic fluid 73 ||| 1 0.00108018 1 0.0687977 2.718
、 この 流体 ||| , each fluid ||| 1 0.00229816 1 0.000525922 2.718

句対応の例：自動車 を含む句の例

、 「 自動車 ||| , ‘‘ automobile ||| 1 0.142122 1 0.0659882 2.718
、 「 自動車 電話 ||| , ‘‘ automobile telephone ||| 1 0.0746708 1 0.0524542 2.718
、 「 自動車 電話 」 ||| , ‘‘ automobile telephone ’’ ||| 1 0.0258531 1 0.0386858 2.718
、 １０２ は 自動車 用 ||| , 102 is a vehicle ||| 1 0.000161113 1 0.00350135 2.718
、 １０２ は 自動車 用 エンジン ||| , 102 is a vehicle engine ||| 1 9.82163e-05 1 0.0031887 2.718
、 ３ は 自動車 ||| , 3 a mobile ||| 1 0.000371807 1 0.000168593 2.718
、 ３ は 自動車 電話 ||| , 3 a mobile telephone ||| 1 0.000195348 1 0.000134015 2.718
、 ３ は 自動車 電話 局 ||| , 3 a mobile telephone office ||| 1 7.2785e-05 1 3.45847e-06 2.718
、 ３ は 自動車 電話 局 、 ||| , 3 a mobile telephone office , ||| 1 4.96815e-05 1 1.31658e-06 2.718
、 ７ は 自動車 ||| , 7 a mobile ||| 1 0.000370597 1 0.000217899 2.718
、 ７ は 自動車 電話機 ||| , 7 a mobile telephone equipment ||| 1 5.71506e-05 1 4.29458e-07 2.718
、 ７ は 自動車 電話機 、 ||| , 7 a mobile telephone equipment , ||| 1 3.90098e-05 1 1.63488e-07 2.718
、 ＥＣ セル ６ を通じて 自動車 ||| vehicle through the ec cell 6 ||| 1 0.00011244 1 0.0167565 2.718
、 ＯＡ 機器 、 自動車 ||| , business machines , automobiles ||| 1 3.50735e-05 1 2.53658e-07 2.718
、 ＯＡ 機器 、 自動車 、 ||| , business machines , automobiles and ||| 1 4.58492e-06 1 9.529e-09 2.718
、 ＯＡ 機器 、 自動車 、 精密 ||| , business machines , automobiles and precision ||| 1 3.50085e-07 1 2
、 この よう な 電気 自動車 ||| , such an electric vehicle ||| 0.5 1.71207e-05 1 3.34838e-05 2.718
、 この よう な 電気 自動車 の ||| , such an electric vehicle ||| 0.5 4.51596e-06 1 3.34838e-05 2.718
、 この ハイブリッド 自動車 ||| , in the hybrid vehicle ||| 1 0.000761246 0.5 0.00269411 2.718
、 この ハイブリッド 自動車 ||| , in this hybrid vehicle ||| 1 0.0064154 0.5 0.00182924 2.718
、 この ハイブリッド 自動車 ５００ ||| , in the hybrid vehicle 500 ||| 1 0.00071914 1 0.0022104 2.718
、 この ハイブリッド 自動車 ５００ において ||| , in the hybrid vehicle 500 , ||| 1 2.0926e-05 1 0.000841
、 この ハイブリッド 自動車 ６００ ||| , in this hybrid vehicle 600 ||| 1 0.00591218 1 0.00136645 2.718
、 この ハイブリッド 自動車 ６００ において ||| , in this hybrid vehicle 600 , ||| 1 0.000172036 1 0.0005
、 しかも 自動車 ||| the automotive ||| 0.1 0.000244648 1 0.0106571 2.718
、 たとえば 自動車 用 ||| , for example , an automobile ||| 0.4 0.000664619 1 0.000540882 2.718
、 たとえば 自動車 用 エンジン ||| , for example , an automobile engine ||| 1 0.000405159 1 0.000492584
、 コードレス 電話 、 自動車 ||| , cordless telephones , car ||| 1 0.0338496 1 0.000443696 2.718
、 コードレス 電話 、 自動車 電話 ||| , cordless telephones , car telephones ||| 1 0.0157655 1 1.50196e-
、 ディーゼルエンジン 自動車 等 ||| diesel engine automobiles and the like ||| 1 0.0151152 1 3.01743e-05
、 ディーゼルエンジン 自動車 等 に ||| on diesel engine automobiles and the like ||| 1 0.00642812 1 2.24
、 ディジタル 自動車 電話 ||| a digital cellular ||| 0.2 2.39509e-05 1 0.000163602 2.718
、 ディジタル 自動車 電話 システム ||| a digital cellular system ||| 1 9.71561e-06 1 0.000148973 2.718
、 デイジタル 自動車 ||| the digital cellular ||| 1 5.80871e-05 1 0.00122759 2.718
、 デイジタル 自動車 電話 ||| the digital cellular telephone ||| 1 3.05191e-05 1 0.000975814 2.718
、 デイジタル 自動車 電話 ５０ ||| the digital cellular telephone 50 ||| 1 2.83228e-05 1 0.000844553 2.7
、 デイジタル 自動車 電話 ５０ において ||| the digital cellular telephone 50 , ||| 1 8.24154e-07 1 0.000
、 ハイブリッド 自動車 ||| a hybrid powered automobile ||| 0.5 0.0354085 0.333333 8.32655e-05 2.718
、 ハイブリッド 自動車 ||| the hybrid car ||| 0.166667 0.0441317 0.333333 0.0427639 2.718
、 ハイブリッド 自動車 ||| the hybrid vehicle ||| 0.0263158 0.0102417 0.333333 0.0694053 2.718
、 ハイブリッド 自動車 ５００ ||| the hybrid vehicle 500 ||| 1 0.00967523 1 0.056944 2.718
、 ハイブリッド 自動車 ５００ の ||| of the hybrid vehicle 500 ||| 1 0.00706622 1 0.0220125 2.718
、 ハイブリッド 自動車 ５００ の ヨー 方向 ||| yawing directions of the hybrid vehicle 500 ||| 1 0.00101
、 ハイブリッド 自動車 の ||| the hybrid car ||| 0.166667 0.0116407 1 0.0427639 2.718
、 ファクシミリ 、 自動車 ||| , facsimiles , automotive vehicles ||| 1 0.0560437 1 1.15431e-06 2.718
、 プラスチック 廃棄 物 、 自動車 ||| , plastic wastes , and automobile ||| 1 0.00690582 1 1.86092e-05 2

現行の句対応抽出の問題点

- 発見的な方法に基づいているため，改良がむずかしい

  発見的の意味：抜き出された句対応が良い句対応かどうかの評価が，機械翻訳実験による性能の変化でしか測定できない．確率的なモデルがない．確率的なモデルがあれば，最尤となる句対応を選べば，そのモデルの観点からは最良の句対応が得られる．

- しかし,「こういう句対応が良いのでは」といういくつかのモデルは，この発見的な方法と同等か少し性能が落ちる．

句の良さの評価

句単位に翻訳していく方法では，なるべく良い句対応を使った翻訳文を得たいので，そのためのスコアを定義する．一つの句対応について次の5つのスコアが良く使われる．

- 句翻訳確率 $\phi(\overline{f}|\overline{e})$
- 句翻訳確率 $\phi(\overline{e}|\overline{f})$
- 単語確率より $\mathrm{lex}(\overline{f}|\overline{e})$
- 単語確率より $\mathrm{lex}(\overline{e}|\overline{f})$
- 句ペナルティ $\exp(1)$ (固定値)

句翻訳確率 $\phi(\overline{f}|\overline{e})$

$$\phi(\overline{f}|\overline{e}) = \frac{\mathrm{count}(\overline{f}, \overline{e})}{\Sigma_{\overline{f}'}\,\mathrm{count}(\overline{f}', \overline{e})}$$

$$\mathrm{count}(\overline{f}, \overline{e}) = \text{句}\overline{f}\text{と}\overline{e}\text{が対応した回数}$$

$$\sum_{\overline{f}'}\mathrm{count}(\overline{f}', \overline{e}) = \text{句}\overline{e}\text{が出現した回数}$$

$\phi(\overline{f}|\overline{e})$は，句$\overline{e}$が出現したとき，それが$\overline{f}$に翻訳される確率と考えられる．$\phi(\overline{e}|\overline{f})$も同様である．

単語確率より $\mathbf{lex}(\overline{f}|\overline{e})$

$$\mathrm{lex}(\overline{f}|\overline{e}) = \max_{\mathbf{a}} P_w(\overline{f}|\overline{e}, \mathbf{a})$$

$$P_w(\overline{f}|\overline{e}, \mathbf{a}) = \prod_{i=1}^{n} E_w(f_i|\overline{e}, \mathbf{a})$$

$$E_w(f_i|\overline{e}, \mathbf{a}) = \frac{1}{\{j|(i,j) \in \mathbf{a}\}} \sum_{(i,j)\in\mathbf{a}} w(f_i|e_j)$$

$$w(f_i|e_j) = \text{単語対応の確率} = \frac{\mathrm{count}(f_i, e_j)}{\Sigma_{f'}\,\mathrm{count}(f', e_j)}$$

$E_w(f_i|\overline{e}, \mathbf{a}) = w(f_i|e_j)$の$e_j$に関する平均値

$P_w(\overline{f}|\overline{e}, \mathbf{a}) = E_w(f_i|\overline{e}, \mathbf{a})$の積. f_iの条件付き独立を仮定

$\mathrm{lex}(\overline{f}|\overline{e}) = P_w(\overline{f}|\overline{e}, \mathbf{a})$が最大の$\mathbf{a}$に関する確率を採用

句ペナルティ

$\exp(1)$に固定．スコアとしては，対数を利用するので，$\log(\exp(1)) = 1$となる．
翻訳に使われた句の$\log$(句ペナルティ)の和は，翻訳に使われた句の数である．
実際の翻訳のスコアには，この句ペナルティ(翻訳に使われた句の数)に，ある重みλを掛けたものを利用する．
このλは自動推定する．
もし，$\lambda > 0$なら，句の数が多いほど翻訳候補のスコアは良くなる．
$\lambda < 0$なら，句の数が少ないほど良い．
通常は $\lambda < 0$で，少ない方が良い．これは同じ単語数からなる文であれば，使われた句の数が少ない (句の長さは長い)方が良いことになる．

フレーズ抽出まとめ

発見的手続きを使うことにより，日英のフレーズ対応を得て，そこに各種のスコアを付けることができることをみた.

問題 (15分)

自動作成したフレーズテーブルには，さまざまな誤りが含まれる．そのようなフレーズテーブルの中から正しそうなフレーズのみを自動抽出するには，どうしたら良いか．

回答例1

	$\overline{f}$	$\overline{f}$ 以外
$\overline{e}$	a	b
$\overline{e}$以外	c	d

$$
\begin{aligned}
a &= \overline{e}と\overline{f}の対応数 \\
b &= \overline{e}の出現数 - a \\
c &= \overline{f}の出現数 - a \\
d &= 句の対応の総数 - a - b - c
\end{aligned}
$$

とすると $\frac{a}{c} \gg \frac{b}{d}$ のとき，$\overline{e}$ と $\overline{f}$ は良い対応と考えられる．これは，$\overline{e}$ と $\overline{f}$ の独立性の検定をして，独立性の低いものが良いということなので，その検定結果により，独立性の低い順に句対応をソートして，上位のみを利用する．

回答例2

- パラレルコーパスを2つに分ける.
- それぞれのコーパスから1つずつフレーズテーブルを作る
- 両方のフレーズテーブルに含まれるようなフレーズのみを採用する

これらの回答例の得失

得 ある程度は信頼性の高いフレーズテーブルができる

失 カバレッジが低くなる

→ 得が大きいか，失が大きいかは実験してみないと分からない.
今のところ，翻訳の性能を下げずに，ノイズ除去ができるらしいということが分かっている.

16. 対数線形モデルの導入

内山将夫@NICT
mutiyama@nict.go.jp

翻訳候補へのスコア付け

入力文を $\mathbf{f}$
翻訳候補を $\mathbf{e}$
$\mathbf{e}$ のスコアを $S_{\mathbf{f}}(\mathbf{e})$ とする.
翻訳システム (デコーダー) は

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} S_{\mathbf{f}}(\mathbf{e})$$

なる $\hat{\mathbf{e}}$ を出力する.
$\arg\max$ の部分は，デコーダーが最大スコアであるような $S_{\mathbf{f}}(\mathbf{e})$ を探す部分である.
以下では，スコア付けの方法について述べる.

確率に基づくスコア付け

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{f})$$

つまり，入力$\mathbf{f}$に対して，最大確率となる$\hat{\mathbf{e}}$を出力するのが基本である．その確率の定式化として

1. ベイズの定理に基づく方法
2. 対数線形モデルに基づく方法

の2通りがある．
2は1の一般化であり，現状では，多くのSMTシステムがこの方法を利用している．まず，1から説明する．

ベイズの定理に基づく方法

$$\begin{aligned}\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) \\ &= \arg\max_{\mathbf{e}} P(\mathbf{f}|\mathbf{e})P(\mathbf{e})\end{aligned}$$

つまり $P(\mathbf{e}|\mathbf{f})$ のかわりに $P(\mathbf{f}|\mathbf{e})P(\mathbf{e})$ を計算する．ここで

- $P(\mathbf{e}) =$ 言語モデルによる $\mathbf{e}$ の確率．つまり，生成された $\mathbf{e}$ のつくりの良さを確率で表現したもの
- $P(\mathbf{f}|\mathbf{e}) = \mathbf{e}$ を条件としたときの $\mathbf{f}$ の確率．$\mathbf{f}$ の翻訳としての良さを確率で表現したもの

これについては,「初歩の確率」のときに少し言及してある．

ベイズの定理と現実との乖離

ベイズの定理の式自体は，正しい式である．しかし，現実には，$P(\cdot)$ の推定に様々な誤りが含まれるため，ベイズの定理に基づく方法は，現実には最適な方法ではない．
$P(\cdot)$ の推定における誤りには，以下のものがある．

- モデルの誤り
- パラメタ推定の誤り

モデルの誤りとは，対象の確率モデルを作るときに，そのモデル自体に誤りがあるため，たとえそのモデルに従って完全なパラメタ推定ができたとしても，対象を正確に表現できないことを言う．実際には，言語を正確に確率モデルとして表現したものは存在しないので，全てのモデルには，程度の大小があったとしても，モデルの誤りがある．
パラメタ推定の誤りとは，与えられたモデルに従って，与えられたデータからモデルのパラメタを推定したときに，データの性質やパラメタ推定法等により，パラメタ推定が上手くいかないことである．パラメタ推定の誤りも必ず存在する．

ベイズの定理に基づく方法でできないこと

1. 翻訳モデルと言語モデルの重みを変えることができない．しかし
$$P(\mathbf{f}|\mathbf{e})^{\lambda_1} P(\mathbf{e})^{\lambda_2}$$
のように重み付けをすることにより，翻訳精度が高くなることがわかっている．
2. 色々な情報を入れることができない．たとえば，$P(\mathbf{f}|\mathbf{e})$ だけでなく，$P(\mathbf{e}|\mathbf{f})$ を追加して，
$$P(\mathbf{f}|\mathbf{e})^{\lambda_1} P(\mathbf{e})^{\lambda_2} P(\mathbf{e}|\mathbf{f})^{\lambda_3}$$
のようにしたい．

上記のようなことをするためには，対数線形モデルが便利である．

ベイズの定理から対数線形モデルに

$$P(\mathbf{e}|\mathbf{f}) = \frac{P(\mathbf{f}|\mathbf{e})P(\mathbf{e})}{P(\mathbf{f})} = \frac{P(\mathbf{f}|\mathbf{e})P(\mathbf{e})}{\Sigma_{\mathbf{e}'} P(\mathbf{f}|\mathbf{e}')P(\mathbf{e}')}$$

$$\begin{aligned} P(\mathbf{f}|\mathbf{e})P(\mathbf{e}) &= \exp(\log P(\mathbf{f}|\mathbf{e}) + \log P(\mathbf{e})) \\ &= \exp(\lambda_1 \log P(\mathbf{f}|\mathbf{e}) + \lambda_2 \log P(\mathbf{e})) \\ &= \exp(\lambda_1 h_1(\mathbf{f},\mathbf{e}) + \lambda_2 h_2(\mathbf{f},\mathbf{e})) \\ &= \exp(\sum_{i=1}^{2} \lambda_i h_i(\mathbf{f},\mathbf{e})) \end{aligned}$$

ただし，

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 1 \\ h_1(\mathbf{f},\mathbf{e}) &= \log P(\mathbf{f}|\mathbf{e}) \\ h_2(\mathbf{f},\mathbf{e}) &= \log P(\mathbf{e}) \end{aligned}$$

よって，

$$P(\mathbf{e}|\mathbf{f}) = \frac{\exp(\Sigma_{i=1}^{2} \lambda_i h_i(\mathbf{f},\mathbf{e}))}{\Sigma_{\mathbf{e}'} \exp(\Sigma_{i=1}^{2} \lambda_i h_i(\mathbf{f},\mathbf{e}'))}$$

一般化

M 個の関数 $h_i(\mathbf{e}, \mathbf{f})$
M 個の重み λ_i について

$$\begin{aligned}
\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) \\
&= \arg\max_{\mathbf{e}} \frac{\exp\left(\Sigma_{i=1}^{M} \lambda_i h_i(\mathbf{f}, \mathbf{e})\right)}{\Sigma_{\mathbf{e}'} \exp\left(\Sigma_{i=1}^{M} \lambda_i h_i(\mathbf{f}, \mathbf{e}')\right)} \\
&= \arg\max_{\mathbf{e}} \sum_{i=1}^{M} \lambda_i h_i(\mathbf{f}, \mathbf{e}) \qquad (1)
\end{aligned}$$

ようするに，各関数 $h_i(\mathbf{e}, \mathbf{f})$ の重み付きの和をもってスコアとする．
この関数を素性という．
対数線形モデルを利用することにより，様々な素性をスコアとして考慮できる．この素性は良く考えて決める．
たとえば，フレーズテーブルのスコアが素性のスコアとなる．
重みは，後述の手法により，自動的に決める．

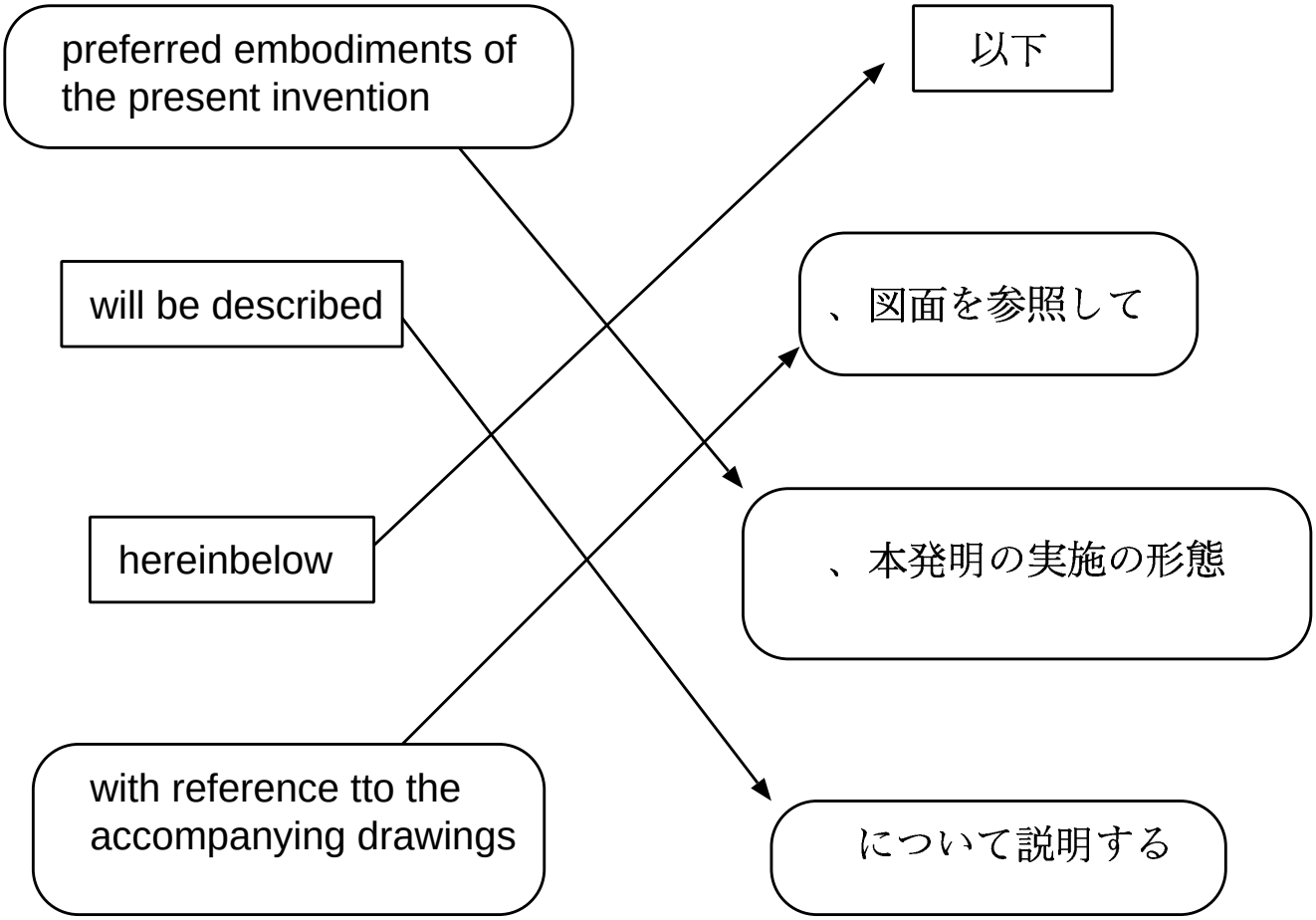
17. 句に基づく統計的機械翻訳(SMT)における素性

内山将夫@NICT
mutiyama@nict.go.jp

句に基づく統計的機械翻訳 (SMT) における素性

- 句単位の翻訳の例
- 句単位の翻訳を構成する基本要素
- 句単位の翻訳の素性

句単位の英日翻訳の例



- 英語の文を句にわける
  (e.g., will be described)
- 各句を翻訳する
  (e.g., について説明する)
- 句を並べ替える

句単位の翻訳の基本要素

- 英語の文$\mathbf{e}$を句にわける

$$\mathbf{e} = \overline{e}_1 \overline{e}_2 \ldots \overline{e}_l$$

- 各句$\overline{e}_i$を日本語の句$\overline{n}_i$に翻訳する

$$\mathbf{n} = \overline{n}_1 \overline{n}_2 \ldots \overline{n}_l$$

- $\overline{n}_i$を並べかえる.

句単位の翻訳のための素性

原言語を $\mathbf{f}$，対象言語を $\mathbf{e}$ としたとき，

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \sum_i \lambda_i h_i(\mathbf{e}, \mathbf{f})$$

なる $\hat{\mathbf{e}}$ を探したい．そのために，$\mathbf{e}$ と $\mathbf{f}$ の組をスコア付けするために，素性 $h_i(\mathbf{e}, \mathbf{f})$ を利用する．

素性の例

- 言語モデル
- 句単位の翻訳モデル
- 句を構成する単語に基づく翻訳モデル
- 単語ペナルティ
- 句ペナルティ
- 語順ペナルティ

これらの素性の値は，$\mathbf{e}, \mathbf{f}$ より $h_i(\mathbf{e}, \mathbf{f})$ として計算される．各素性の重みは，自動推定する (後述).

言語モデル

$$
\begin{aligned}
h(\mathbf{e}, \mathbf{f}) &= \log P(\mathbf{e}) \\
&= \log P(e_1, e_2, \ldots, e_l) \\
&= \log \prod_i P(e_i | e_{i-2}, e_{i-1})
\end{aligned}
$$

3-gram 言語モデルにより対象言語 $\mathbf{e}$ の生成確率の対数を素性とする．これが大きい翻訳文 $\mathbf{e}$ は，対象言語としてのつくりが良いと考えられる．

ところで，

3-gram 言語モデルは，前の 2 単語しか見ていないので，とても貧弱なモデルに見えるが，これを越えるモデルはあまりない．良い言語モデルを作ることができれば，機械翻訳に与えるインパクトは大きい．

単語ペナルティと句ペナルティ

単語ペナルティ

$$h(\mathbf{e}, \mathbf{f}) = \mathbf{e}\text{中の単語数}$$

句ペナルティ

$$h(\mathbf{e}, \mathbf{f}) = \mathbf{e}\text{中の句数}$$

これらの重みが正のときには，単語や句がたくさんある$\mathbf{e}$が優先される．負のときには，単語や句は少ない方が良い．
単語ペナルティや句ペナルティは，適切な長さの訳文を出力するために有用である．

句単位の翻訳モデル

$$\begin{aligned} h(\mathbf{e},\mathbf{f}) &= \log P(\mathbf{e}|\mathbf{f}) \\ &= \log \Pi P(\overline{e}|\overline{f}) \\ &= \Sigma \log P(\overline{e}|\overline{f}) \end{aligned}$$

これは $\mathbf{f} \rightarrow \mathbf{e}$ の翻訳モデルだが，逆方向も同様に，

$$h(\mathbf{e},\mathbf{f}) = \Sigma \log P(\overline{f}|\overline{e})$$

ただし，

$$\begin{aligned} P(\overline{f}|\overline{e}) &= \frac{\mathrm{count}(\overline{f},\overline{e})}{\Sigma_{\overline{f}'}\,\mathrm{count}(\overline{f}',\overline{e})} \\ \mathrm{count}(\overline{f},\overline{e}) &= \text{句}\overline{f}\text{と句}\overline{e}\text{が対応している回数} \end{aligned} \quad (1)$$

$P(\overline{f}|\overline{e})$ の問題点

$\mathrm{count}(\overline{f},\overline{e})$ が小さいときに値が信頼できない

$\rightarrow$ スムージング

単語に基づく句対応の重み (双方向)

$$h(\mathbf{e}, \mathbf{f}) = \sum \log \mathrm{lex}(\overline{f}|\overline{e}) \tag{2}$$

$$\mathrm{lex}(\overline{f}|\overline{e}) = \max_{\mathbf{a}} P_w(\overline{f}|\overline{e}, \mathbf{a})$$

$$P_w(\overline{f}|\overline{e}, \mathbf{a}) = \prod_{i=1}^{n} E_w(f_i|\overline{e}, \mathbf{a})$$

$$E_w(f_i|\overline{e}, \mathbf{a}) = \frac{1}{\{j|(i,j) \in \mathbf{a}\}} \sum_{(i,j)\in\mathbf{a}} w(f_i|e_j)$$

$$w(f_i|e_j) = \text{単語対応の確率} = \frac{\mathrm{count}(f_i, e_j)}{\Sigma_{f'} \mathrm{count}(f', e_j)}$$

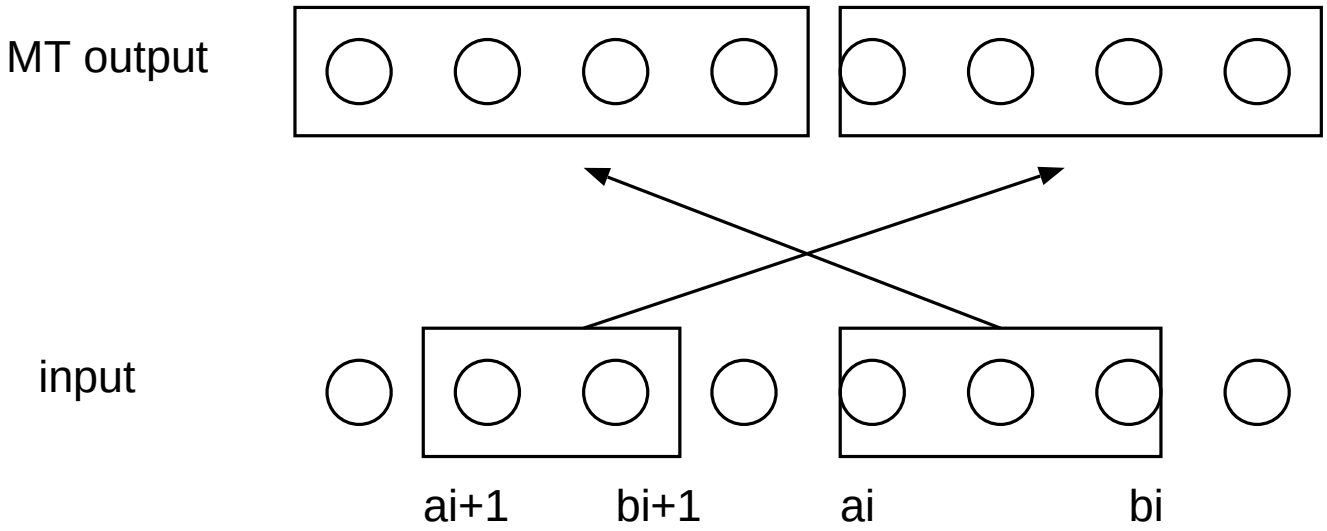
$E_w(f_i|\overline{e}, \mathbf{a}) = w(f_i|e_j)$の$e_j$に関する平均値

$P_w(\overline{f}|\overline{e}, \mathbf{a}) = E_w(f_i|\overline{e}, \mathbf{a})$の積. f_iの条件付き独立を仮定

$\mathrm{lex}(\overline{f}|\overline{e}) = P_w(\overline{f}|\overline{e}, \mathbf{a})$が最大の$\mathbf{a}$に関する確率を採用

語順

語順については，今，盛んに研究されている．単純なものとしては，原言語の語順と異なるものにペナルティを与えるというものがある．



のとき，

$$|b_i - a_{i+1} + 1| \;=\; |\overline{f}_i\text{に対応する}\overline{e}_i\text{の最後の単語位置} \\ -\; \overline{f}_{i+1}\text{に対応す}\overline{e}_{i+1}\text{の最初の単語位置} + 1|$$

として，連続するフレーズが連続するときのペナルティを 0 とし，そうでないときに，離れ具合に応じてペナルティをかける．

まとめ

- 対数線形モデルを利用することにより，様々な素性をスコア付けに利用できる
- 各素性は，それぞれ別個に改良できる

特に，

- 言語モデルの改良
- 翻訳モデルの改良
  - カバー率の高い句表をつくる
  - 良いスコア付けをする
- 語順の制約

が重要である．

18. デコーダ概要

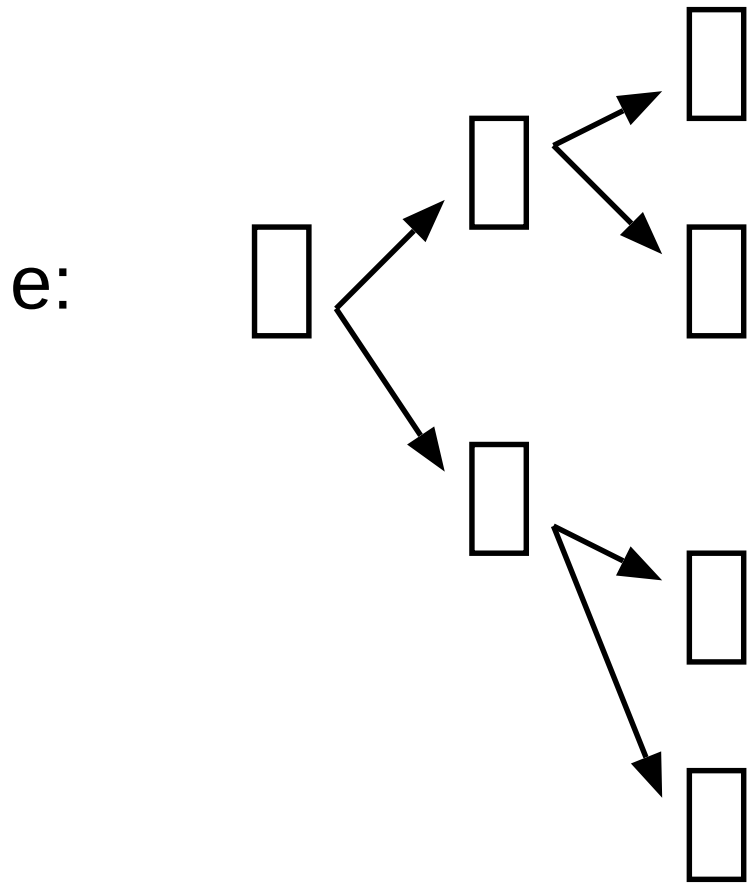
内山将夫@NICT
mutiyama@nict.go.jp

デコーダの役割

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \sum_i \lambda_i h_i(\mathbf{e}, \mathbf{f})$$

における $\hat{\mathbf{e}}$ の探索.

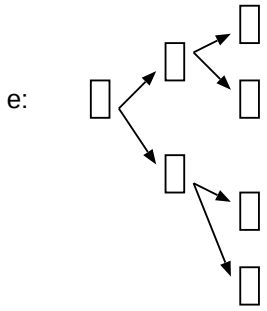
$\mathbf{f}$ を部分的に訳していって，いくつもの仮説を作り，そこから最適なものを得る.



デコーダの基本要素

仮説 (部分翻訳) を生成し，不要なものを消し，良いものを残す.

- 仮説の生成 (部分翻訳の生成)

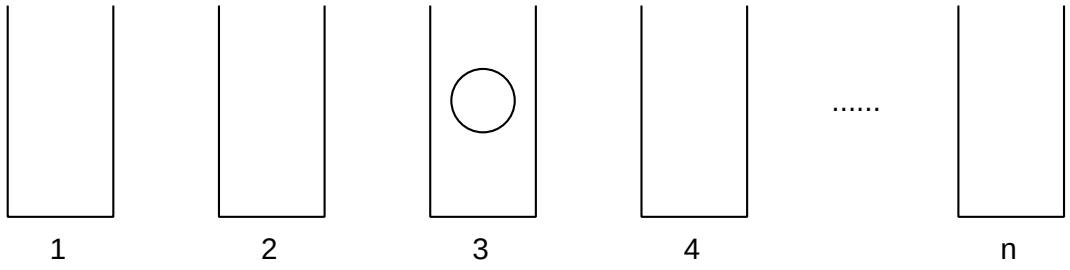


- 仮説の枝刈り (不要な仮説を消す)
- 最も良い仮説の探索

Multi-Stack Beam-Search

$$\mathbf{f} = f_1 f_2 f_3 \ldots f_n$$

とすると，これまでに翻訳された $\mathbf{f}$ の単語数に応じて n 個のスタックをつくる．



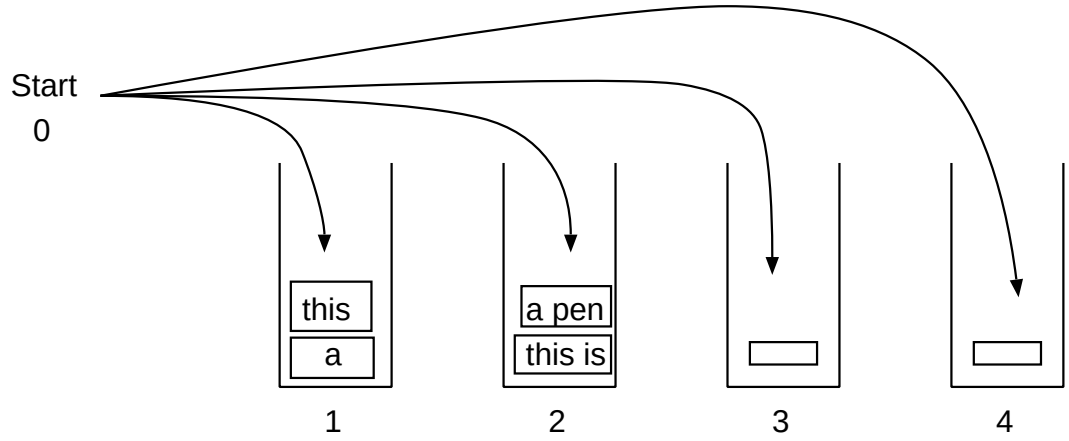
3番目のスタックには，$\mathbf{f}$ の単語を３つ訳した結果の部分翻訳を示す仮説が積まれる．たとえば，

$$\mathbf{f} = f_1 \overline{f}_2 f_3 \overline{f}_4 \overline{f}_5 f_6$$

では，$f_2,\ f_4,\ f_5$ を翻訳した部分に相当する仮説が積まれる．

初期仮説の積み方

初期仮説(カバー0)から，最初の１つのフレーズ翻訳によりカバーされるfの側の単語数に応じて仮説をスタックに分配する．



たとえば,「this is a pen .」を翻訳するときに，フレーズの候補として

- 「this」→「これ」「ここ」...
- 「this is」→「これ は」「それ を」...
- 「a」→「し た」「で は」...
- 「a pen」→「、 ペン」...

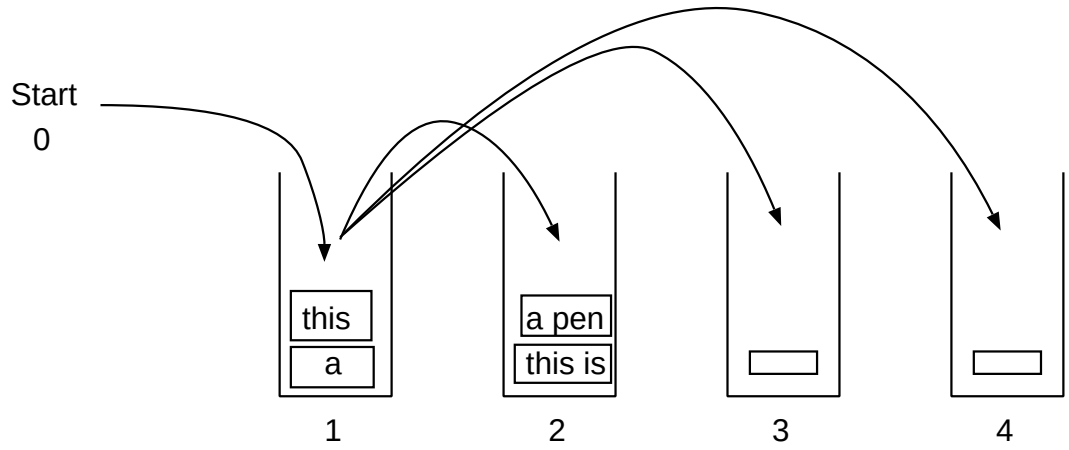
などがある．このとき，

- 「this」と「a」は，カバー1のスタックに載り
- 「this is」と「a pen」は，カバー2のスタックに載る

その他，全てのフレーズについて，そのフレーズがカバーする単語数に応じて，スタックに分配する．

カバー1のスタックからの仮説の展開

カバー1のスタック中の各仮説について，その仮説に後続可能なフレーズを翻訳して，その結果としてカバーした単語数に相当するスタックに載せる．



たとえば,「this」のあとに

- 「is」を翻訳すると，その結果はカバー2のスタックに
- 「a」を翻訳すると，その結果はカバー2のスタックに
- 「a pen」を翻訳すると，カバー3のスタックに

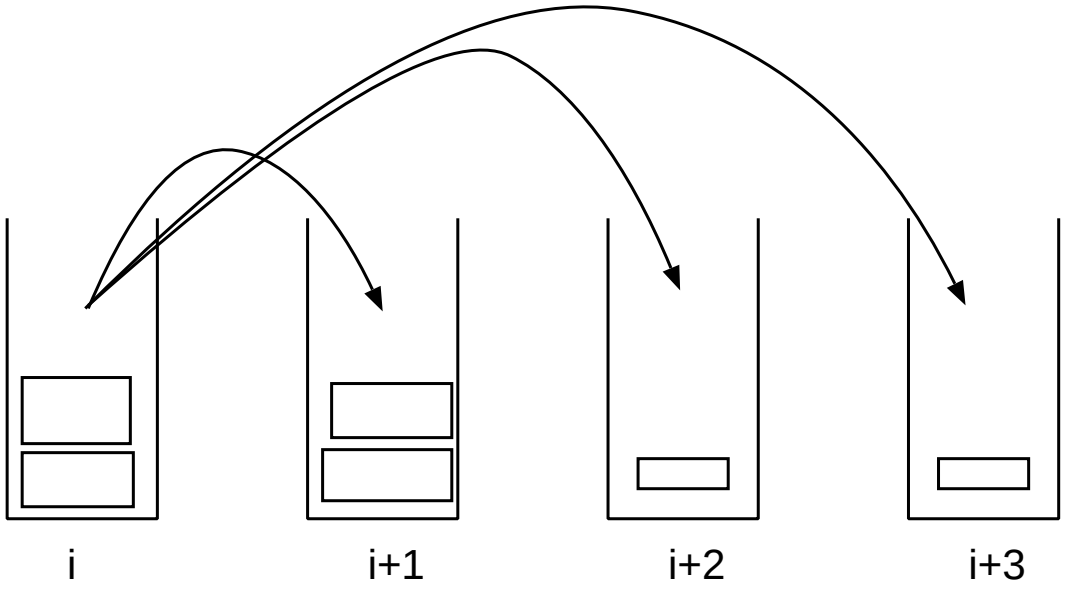
積まれる．その結果として，それぞれ,「this is」「this a」「this a pen」とその訳が積まれている．また,
「a」のあとに

- 「this」を翻訳すると，カバー2のスタックに
- 「is」を翻訳すると，カバー2のスタックに
- 「this is」を翻訳すると，カバー3のスタックに

積まれる．その結果として，それぞれ,「a this」「a is」「a this is」とその訳が積まれている．

カバー2,3,4...iと進む

カバーiのスタックにおいて，入力文から，次の未翻訳のフレーズを翻訳し，そのフレーズを，カバーiのスタック中の部分翻訳に追加した結果の翻訳を，その翻訳がカバーする単語数に対応するスタックに積む.



以上を $1 \leq i \leq n$ まで繰り返すと，スタックnを処理したときには，$f_1, f_2, \ldots, f_n$の全ての単語が翻訳されている.

枝刈りの必要性

1単語は1単語にしか訳されないとしても，これまでの方法では，

- 最初のスタックにn単語
- 次のスタックに$n-1$単語

  …

となって，$n!$の仮説ができる．フレーズを考えるともっと多くなる．
→ 仮説の枝刈りをしないといけない．

スタックに積む仮説：翻訳オプション

- 仮説のそれぞれは，それぞれが1つの E のフレーズ$\overline{f}$をもつ
- このフレーズに対応する E のフレーズ$\overline{e}$ももつ
- 翻訳オプションは，以下の4つの要素を持つ
  - $\overline{f}$の**f**中での開始位置
  - $\overline{f}$の**f**中での終了位置
  - $\overline{e}$
  - フレーズを翻訳するときのコスト

    $$= -\Sigma \lambda_i f_i(\overline{e}, \overline{f})$$

    フレーズテーブルにあるフレーズスコアの重み付きの和 $\times -1$

翻訳オプションは，**f**中の全てのフレーズについて，あらかじめ作っておく.

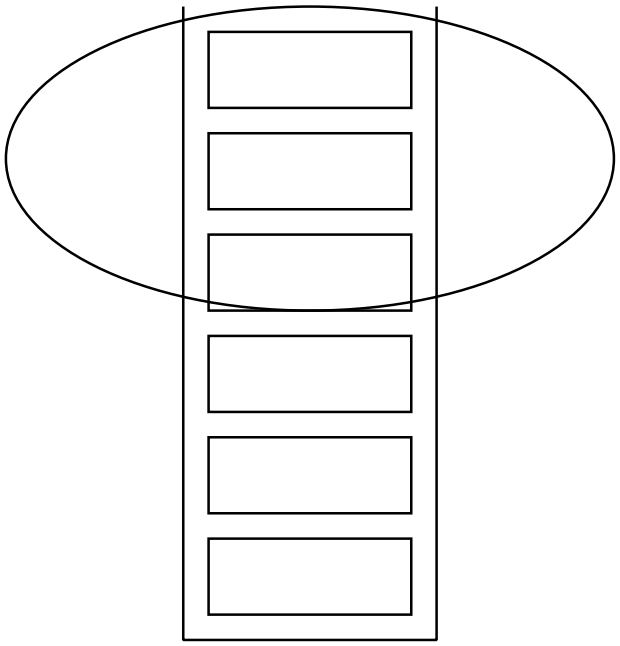
仮説の構成物

- 翻訳オプション
- 1つ前の仮説へのリンク．(これを辿っていけば，どのフレーズがどのフレーズに訳されたかがわかるため，翻訳文を生成できる)
- これまでにカバーされた $\mathbf{f}$ の単語の位置 (重複して同一の単語を訳すことを防ぐため)
- 2つ前までの $\mathbf{e}$ の単語．(3-gram 言語モデルの計算に必要)
- これまでのコストの和
- 予測コスト

コスト最小の仮説を探す.

仮説の枝刈り (Beam Search)

予測全体コストが小さい仮説のみを選んで、
コストが大きいものは捨てる



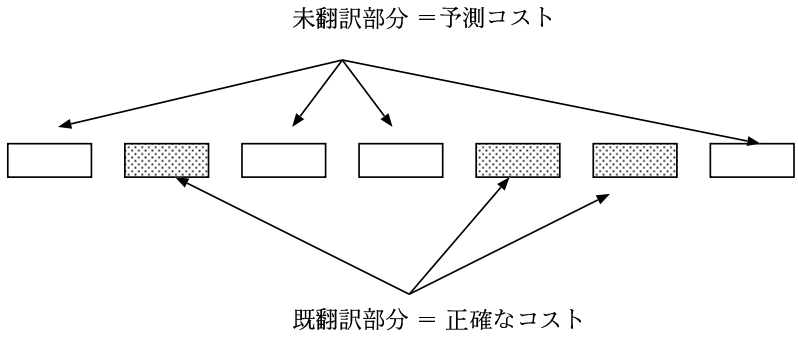
予測全体コスト
＝これまでのコスト
＋これからの予測コスト

翻訳した f の数が等しい仮説が載る

これまでのコスト
＝ $\sum$ フレーズの翻訳コスト
＋ 翻訳された **e** の言語モデルからのコスト
＋ 語順のコスト

翻訳された **e** の言語モデルからのコスト
$= -\sum \log P(e_i|e_{i-2}, e_{i-1})$

予測全体コスト ＝ 既翻訳のコスト ＋ 未翻訳のコスト

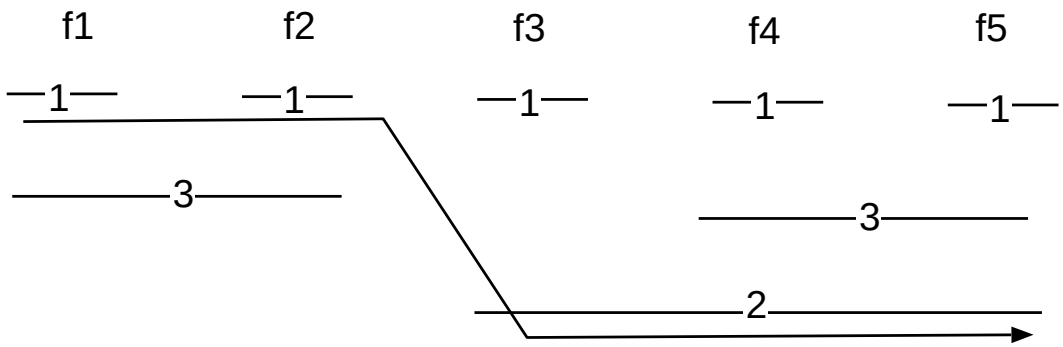


既翻訳部分のコスト ＝ 正確なコスト
＝ $\sum$フレーズの翻訳コスト
＋ 翻訳文の言語モデルコスト
＋ 語順のコスト

未翻訳部分のコスト ＝ 予測コスト
＝ $\sum$フレーズの翻訳コスト
＋ $\sum$フレーズ内言語モデルコスト

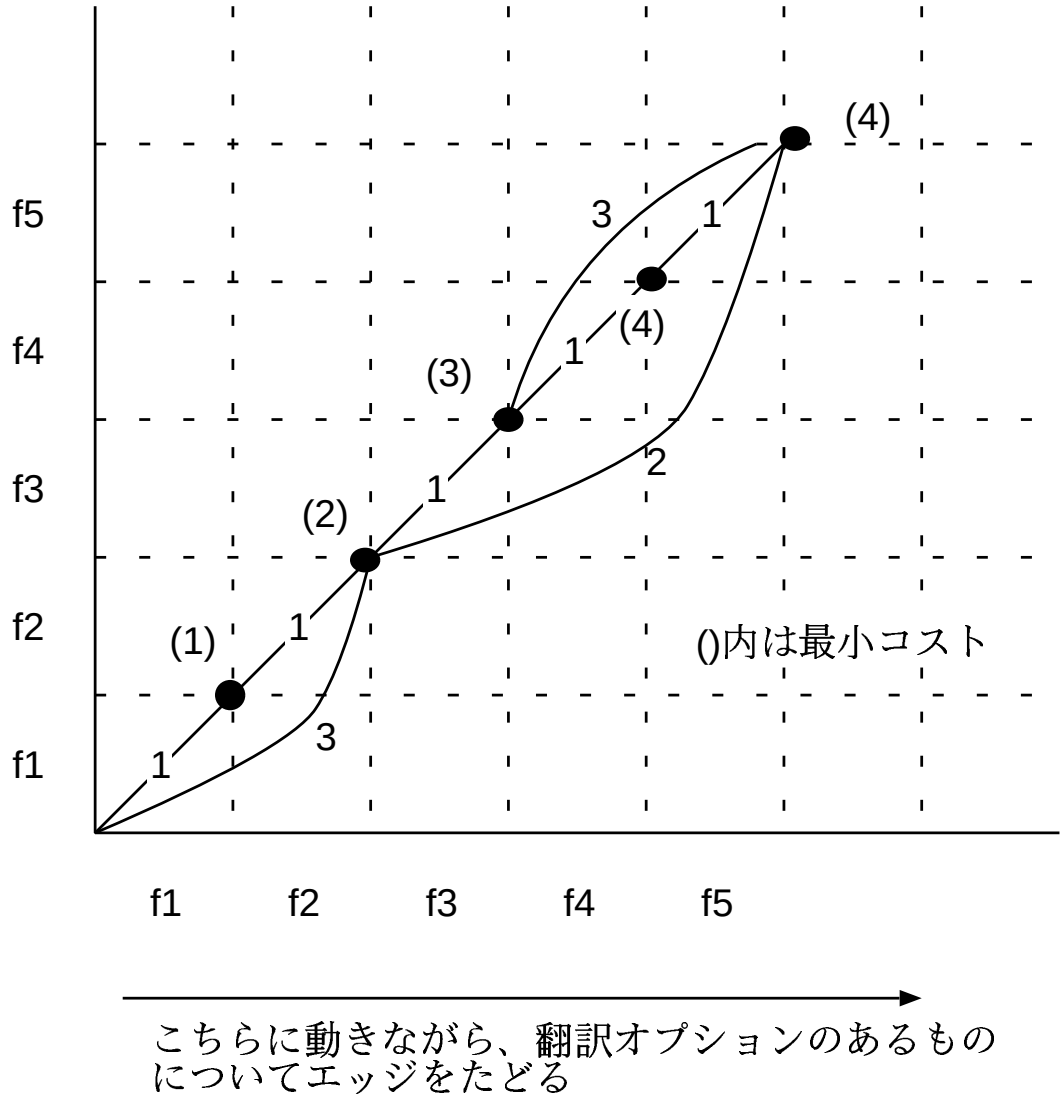
予測コストにおいては，未翻訳のフレーズの翻訳コストの和と未翻訳のフレーズのフレーズ内言語モデルのコストの和を利用する．
フレーズの翻訳コストは，フレーズテーブルにリストされているスコアの和に -1を掛けたものである．
フレーズ内言語モデルコストは，$\bar{e} = e_1 e_2 ... e_n$についての言語モデルによる確率の対数に$-1$を掛けたものである．
フレーズ内言語モデルコストは，フレーズ間の繋ぎ目をまたいだ確率の計算はしない．

予測コストの計算



最小コストパスを通ることにより，翻訳コストを計算する
各フレーズのコストは，そのフレーズに対応する，複数の翻訳の中から最小のコストのものを選ぶ.
したがって，予測コストは，最小のコストとなる.

DPを利用した最小コストパスの計算法



こちらに動きながら、翻訳オプションのあるもの
についてエッジをたどる

実行例

- 入力: this is a pen .
- 出力: これ は 、 ペン で ある 。

翻訳オプション

```
this  ここ，その，て この，この，上記，...
is    さ れ て，に，も，れる，の は，に は，...
a     で は 、，（，し た，Ａ，ａ，...
pen   筆，ペン の，ＰＥＮ，入力 ペン，ｐｅｎ，ペン で，...
.     なる 。，さ れる 。，） 。，し て いる 。，れる 。，...

this is  を，すなわち，は その，それ を，これ が，この こと は，...
is a     が ある，が 、，と は 、，は その，が，の は 、，を 、
a pen    、 ペン，ａ ｐｅｎ ．，た ペン，と 、 ペン，の ペン，...
pen .    ペン で ある 。

this is a  が，この こと が，この こと は 、，これ が，これ が 、，...

is a pen        φ
a pen .         φ
this is a pen   φ
is a pen .      φ
this is a pen . φ

[入力フレーズ;ID]
        翻訳フレーズ<ID>，pC=対数フレーズ翻訳確率，c=コスト

[this;10]
        ここ<1341>，pC=-1.35766，c=-2.1591
        その<1648>，pC=-1.04392，c=-1.67822
        て この<1845>，pC=-1.27158，c=-2.15583
        この<1364>，pC=-0.416247，c=-1.06726
        上記<2943>，pC=-1.61984，c=-2.27863
        の この<2235>，pC=-1.53633，c=-2.30776
        これ<1394>，pC=-0.817653，c=-1.53584
        本<3271>，pC=-0.766849，c=-1.37588
        これら<1402>，pC=-1.6298，c=-2.3598
        以上<2691>，pC=-1.69762，c=-2.21865
        これ に<1396>，pC=-1.50672，c=-2.11998
        に この<2110>，pC=-1.63886，c=-2.28144
        で<239>，pC=-2.04413，c=-2.26692
        は<2315>，pC=-1.93096，c=-2.11451
        で この<1944>，pC=-1.7142，c=-2.32927
        この よう<1373>，pC=-1.35286，c=-2.27915
        以上 の<2692>，pC=-1.91313，c=-2.27875
        それ<1659>，pC=-1.46538，c=-2.1846
        該<2747>，pC=-1.52524，c=-2.26044
        当該<3184>，pC=-1.59398，c=-2.34628

[is;5]
```

さ れ て<3421>, pC=-2.11081, c=-2.02628
に<299>, pC=-1.48134, c=-1.66314
も<344>, pC=-1.47625, c=-1.85074
れる<400>, pC=-1.29528, c=-1.88471
の は<2251>, pC=-1.41678, c=-2.04515
に は<2133>, pC=-1.43534, c=-1.75688
と は<2002>, pC=-1.72335, c=-2.04814
が<1325>, pC=-0.470061, c=-0.647842
の が<2234>, pC=-1.40044, c=-2.0277
が ある<1328>, pC=-1.52732, c=-1.96323
れ<2527>, pC=-1.38216, c=-1.94951
の<2208>, pC=-1.76538, c=-1.90982
を<417>, pC=-0.577587, c=-0.729719
） を<873>, pC=-1.77558, c=-1.93028
） が<847>, pC=-1.68848, c=-1.89245
） は<870>, pC=-1.77264, c=-2.02156
さ れる<154>, pC=-1.8255, c=-2.03365
で は<1951>, pC=-1.51902, c=-1.78469
さ れ<3419>, pC=-1.86897, c=-2.05372
は<2315>, pC=-0.542636, c=-0.726186

[a;3]
で は 、<1952>, pC=-1.99372, c=-1.94916
（<747>, pC=-1.50532, c=-1.732
し た<1473>, pC=-1.4491, c=-1.74456
Ａ<1049>, pC=-0.853386, c=-1.32077
ａ<1121>, pC=-1.09356, c=-1.55888
に は 、<2134>, pC=-1.73294, c=-1.75383
、<8>, pC=-0.684518, c=-0.886027
は<2315>, pC=-0.824683, c=-1.00823
は 、<2316>, pC=-1.20584, c=-1.38524
れ た<2531>, pC=-1.49016, c=-1.97397
た<1666>, pC=-0.745912, c=-1.23491
で 、<1899>, pC=-1.57655, c=-1.89724
な<2033>, pC=-1.18053, c=-1.66961
する<174>, pC=-1.01081, c=-1.39731
さ れ た<1431>, pC=-1.85698, c=-1.80738
て 、<1786>, pC=-1.21214, c=-1.82842
で は<1951>, pC=-1.64188, c=-1.90755
ある<96>, pC=-1.07434, c=-1.72888
に は<2133>, pC=-1.46608, c=-1.78763
で ある<241>, pC=-1.53484, c=-1.77665

[pen;7]
筆<572>, pC=-1.09842, c=-1.91969
ペン の<4625>, pC=-1.00945, c=-1.76537
ＰＥＮ<4601>, pC=-1.16262, c=-1.98389

入力 ペン<4635>, pC=-1.60213, c=-2.38855
ｐｅｎ<4603>, pC=-1.41516, c=-2.23643
ペン で<570>, pC=-1.23726, c=-2.03232
ペン で ある<4623>, pC=-1.60909, c=-2.33354
ペン 筆跡<4628>, pC=-1.47485, c=-2.45062
て ペン<4607>, pC=-0.999509, c=-2.1773
ペン に<4624>, pC=-1.4014, c=-2.30551
で ペン<4609>, pC=-1.37715, c=-2.31839
ペン 入力<4627>, pC=-1.76514, c=-2.49763
に ペン<4611>, pC=-1.3145, c=-2.23882
の ペン<566>, pC=-1.04526, c=-1.9127
入力<4634>, pC=-2.10195, c=-2.58898
ペン ２ の<4620>, pC=-1.61217, c=-2.45588
スタイラス<4614>, pC=-1.67264, c=-2.49391
タッチ<4615>, pC=-1.79274, c=-2.58717
タッチ ペン<4616>, pC=-1.72678, c=-2.5611
ペン<569>, pC=-0.267554, c=-1.04067

[.;1]
なる 。<294>, pC=-1.10377, c=-1.88984
さ れる 。<155>, pC=-1.40193, c=-1.30851
） 。<33>, pC=-1.47075, c=-1.6819
し て いる 。<167>, pC=-1.67653, c=-1.44244
れる 。<401>, pC=-0.996328, c=-1.63918
さ れ て いる 。<150>, pC=-1.91711, c=-1.17936
ある 。<97>, pC=-0.907851, c=-1.72634
する 。<175>, pC=-1.14859, c=-1.62012
て いる 。<208>, pC=-0.942884, c=-1.23263
。<10>, pC=-0.24766, c=-0.713159
もの で ある 。<349>, pC=-1.90485, c=-1.65041
なっ て いる 。<291>, pC=-1.63915, c=-1.49152
と なる 。<275>, pC=-1.72258, c=-1.72477
れ て いる 。<392>, pC=-1.52524, c=-1.34756
よう に なっ て いる 。<368>, pC=-2.508, c=-1.76419
に なっ て いる 。<309>, pC=-2.13272, c=-1.64248
に なる 。<310>, pC=-1.61102, c=-1.75654
と なっ て いる 。<274>, pC=-2.35949, c=-1.75109
られ て いる 。<382>, pC=-2.07856, c=-1.88984
で ある 。<242>, pC=-1.23406, c=-1.14991

[this is;9]
を<417>, pC=-1.54862, c=-1.70076
すなわち<1528>, pC=-1.73591, c=-2.31995
は その<2385>, pC=-1.71699, c=-2.20306
それ を<4688>, pC=-1.64736, c=-2.29564
これ が<4642>, pC=-0.911397, c=-1.62708
この こと は<4666>, pC=-1.43752, c=-2.30474

は これ<2383>, pC=-1.71766, c=-2.34747
ここ で は<1345>, pC=-1.89881, c=-2.25273
が<1325>, pC=-1.50635, c=-1.68413
という の は<4691>, pC=-1.93868, c=-2.18854
それ は<4687>, pC=-1.37946, c=-2.20885
が この<3376>, pC=-1.57157, c=-2.13519
が これ<4657>, pC=-1.5425, c=-2.23305
が これ に<4658>, pC=-1.73319, c=-2.17005
が これ は<4659>, pC=-1.57178, c=-2.24404
は<2315>, pC=-1.45277, c=-1.63632
以上 が<4652>, pC=-1.71494, c=-2.34458
これ を<1401>, pC=-0.962022, c=-1.51385
て これ を<4690>, pC=-1.80236, c=-2.40022
これ は<1399>, pC=-0.652242, c=-1.35359

[is a;4]
が ある<1328>, pC=-1.67738, c=-2.1133
が 、<1326>, pC=-1.28772, c=-1.62281
と は 、<2003>, pC=-1.77381, c=-1.85223
は その<2385>, pC=-1.62439, c=-2.11046
が<1325>, pC=-0.863859, c=-1.04164
の は 、<2252>, pC=-1.68919, c=-2.00883
を 、<2587>, pC=-1.52488, c=-1.88536
は<2315>, pC=-0.660629, c=-0.84418
を<417>, pC=-1.0029, c=-1.15503
） は<870>, pC=-1.94566, c=-2.19457
） は 、<871>, pC=-1.88151, c=-1.8389
の<2208>, pC=-1.9946, c=-2.13903
に は 、<2134>, pC=-1.76211, c=-1.783
で は<1951>, pC=-1.89019, c=-2.15586
に は<2133>, pC=-1.62461, c=-1.94616
で は 、<1952>, pC=-2.01248, c=-1.96793
と は<2002>, pC=-1.87138, c=-2.19618
は 、 この<2324>, pC=-1.91441, c=-2.1825
は 、<2316>, pC=-0.796328, c=-0.975726
に<299>, pC=-1.64566, c=-1.82745

[a pen;2]
、 ペン<562>, pC=-0.753549, c=-1.62573
ａ ｐｅｎ ． <563>, pC=-2.12552, c=-3.23558
た ペン<564>, pC=-0.932183, c=-2.08682
と 、 ペン<565>, pC=-1.4245, c=-2.38424
の ペン<566>, pC=-0.944163, c=-1.8116
は 、 ペン<567>, pC=-1.31232, c=-2.19307
は ペン<568>, pC=-1.16434, c=-2.07313
ペン<569>, pC=-0.44751, c=-1.22063
ペン で<570>, pC=-1.223, c=-2.01806

```
        握り<571>, pC=-1.77062, c=-2.59188
        筆<572>, pC=-1.377, c=-2.19827
        例えば ペン 入力<573>, pC=-1.84151, c=-3.13061

[pen .;6]
        ペン で ある 。<4599>, pC=-0.68607, c=-1.08455

[this is a;8]
        が<1325>, pC=-1.70286, c=-1.88064
        この こと が<4640>, pC=-1.41113, c=-2.60653
        この こと は 、<4641>, pC=-1.55813, c=-2.26904
        これ が<4642>, pC=-1.07722, c=-1.7929
        これ が 、<4643>, pC=-1.31737, c=-1.97721
        これ で は 、<4644>, pC=-1.62747, c=-1.9997
        これ に<1396>, pC=-1.88038, c=-2.49364
        これ は<1399>, pC=-0.923228, c=-1.62458
        これ は 、<4645>, pC=-1.03133, c=-1.43306
        これ は 、 いわゆる<4646>, pC=-1.85256, c=-2.76921
        これ は , <4647>, pC=-1.81499, c=-2.83232
        以上 が<4652>, pC=-1.67579, c=-2.30544
        これ は また 、<4649>, pC=-1.71063, c=-2.51877
        これら が<4650>, pC=-1.64068, c=-2.54958
        すなわち<1528>, pC=-1.87628, c=-2.46032
        ただし 、 これ は<4651>, pC=-1.78216, c=-2.85429
        の<2208>, pC=-2.20365, c=-2.34808
        は<2315>, pC=-1.79407, c=-1.97762
        は 、<2316>, pC=-1.87083, c=-2.05023
        も<344>, pC=-1.92003, c=-2.29452

[is a pen;0]
[a pen .;0]
[this is a pen;0]
[is a pen .;0]
[this is a pen .;0]
```

予測コストの計算

```
4   .  |
3 pen  |
2   a  |
1  is  |
0 this |
       +-------+------+-----+-------+----
        0 this | 1 is | 2 a | 3 pen | 4 .

computing future cost from 0 to 0
        can get to 1 with cost -2.1591
        can get to 1 with cost -1.67822         this
        can get to 1 with cost -1.06726         この
        => cheapest way: -1.06726
computing future cost from 0 to 1
        can get to 1 with cost -2.1591
        can get to 1 with cost -1.67822
        can get to 1 with cost -1.06726
        can get to 2 with cost -1.70076
        can get to 2 with cost -1.62708
        can get to 2 with cost -1.51385         this is
        can get to 2 with cost -1.35359         これ は
        => cheapest way: -1.35359
computing future cost from 0 to 2
        can get to 1 with cost -2.1591
        can get to 1 with cost -1.67822
        can get to 1 with cost -1.06726
        can get to 2 with cost -1.70076
        can get to 2 with cost -1.62708
        can get to 2 with cost -1.51385
        can get to 2 with cost -1.35359
        can get to 3 with cost -1.88064
        can get to 3 with cost -1.7929
        can get to 3 with cost -1.62458         this is a
        can get to 3 with cost -1.43306         これ は 、
        => cheapest way: -1.43306
computing future cost from 0 to 3
        can get to 1 with cost -2.1591
        can get to 1 with cost -1.67822
        can get to 1 with cost -1.06726
        can get to 2 with cost -1.70076
        can get to 2 with cost -1.62708
        can get to 2 with cost -1.51385
        can get to 2 with cost -1.35359
        can get to 3 with cost -1.88064
        can get to 3 with cost -1.7929
```

```
        can get to 3 with cost -1.62458
        can get to 3 with cost -1.43306
        can get to 4 with cost -2.97932
        can get to 4 with cost -2.57422     -1.43306 -1.04067
        can get to 4 with cost -2.47373     this is a + pen
        => cheapest way: -2.47373           これ は 、+ ペン
computing future cost from 0 to 4
        can get to 1 with cost -2.1591
        can get to 1 with cost -1.67822
        can get to 1 with cost -1.06726
        can get to 2 with cost -1.70076
        can get to 2 with cost -1.62708
        can get to 2 with cost -1.51385
        can get to 2 with cost -1.35359
        can get to 3 with cost -1.88064
        can get to 3 with cost -1.7929
        can get to 3 with cost -1.62458
        can get to 3 with cost -1.43306
        can get to 4 with cost -2.97932
        can get to 4 with cost -2.57422
        can get to 4 with cost -2.47373     -1.43306 -1.08455
        can get to 5 with cost -2.51762     this is a + pen .
        => cheapest way: -2.51762           これ は 、+ ペン で ある 。
computing future cost from 1 to 1
        can get to 1 with cost -2.02628
        can get to 1 with cost -1.66314
        can get to 1 with cost -0.647842     is
        => cheapest way: -0.647842           が
computing future cost from 1 to 2
        can get to 1 with cost -2.02628
        can get to 1 with cost -1.66314
        can get to 1 with cost -0.647842
        can get to 2 with cost -2.1133
        can get to 2 with cost -1.62281
        can get to 2 with cost -1.04164
        can get to 2 with cost -0.84418      is a
        => cheapest way: -0.84418            は
computing future cost from 1 to 3
        can get to 1 with cost -2.02628
        can get to 1 with cost -1.66314
        can get to 1 with cost -0.647842
        can get to 2 with cost -2.1133
        can get to 2 with cost -1.62281
        can get to 2 with cost -1.04164
        can get to 2 with cost -0.84418
        can get to 3 with cost -2.27357      -0.84418 -1.04067
        can get to 3 with cost -1.86847      is a + pen
```

```
        => cheapest way: -1.86847           は   + ペン
computing future cost from 1 to 4
        can get to 1 with cost -2.02628
        can get to 1 with cost -1.66314
        can get to 1 with cost -0.647842
        can get to 2 with cost -2.1133
        can get to 2 with cost -1.62281
        can get to 2 with cost -1.04164
        can get to 2 with cost -0.84418
        can get to 3 with cost -2.27357
        can get to 3 with cost -1.86847     -0.84418 -1.08455
        can get to 4 with cost -1.92873     is a + pen .
        => cheapest way: -1.92873           は   + ペン で ある 。
computing future cost from 2 to 2
        can get to 1 with cost -1.94916
        can get to 1 with cost -1.732
        can get to 1 with cost -1.32077
        can get to 1 with cost -0.886027    a
        => cheapest way: -0.886027          、
computing future cost from 2 to 3
        can get to 1 with cost -1.94916
        can get to 1 with cost -1.732
        can get to 1 with cost -1.32077
        can get to 1 with cost -0.886027
        can get to 2 with cost -1.62573
        can get to 2 with cost -1.22063     a pen
        => cheapest way: -1.22063           ペン
computing future cost from 2 to 4
        can get to 1 with cost -1.94916
        can get to 1 with cost -1.732
        can get to 1 with cost -1.32077     (微妙に数値が違う)
        can get to 1 with cost -0.886027    一つ前のフレーズを考慮して
        can get to 2 with cost -1.62573     対象言語の言語モデルのコストを
        can get to 2 with cost -1.22063     計算していると思われる.
        can get to 3 with cost -1.97058     -0.886027 -1.08455
        can get to 3 with cost -1.93378     a  + pen .
        => cheapest way: -1.93378           、 + ペン で ある 。
computing future cost from 3 to 3
        can get to 1 with cost -1.91969
        can get to 1 with cost -1.76537
        can get to 1 with cost -1.04067     pen
        => cheapest way: -1.04067           ペン
computing future cost from 3 to 4
        can get to 1 with cost -1.91969
        can get to 1 with cost -1.76537
        can get to 1 with cost -1.04067
        can get to 2 with cost -1.08455     pen .
```

```
        => cheapest way: -1.08455           ペン で ある 。
computing future cost from 4 to 4
        can get to 1 with cost -1.88984
        can get to 1 with cost -1.30851
        can get to 1 with cost -1.17936     .
        can get to 1 with cost -0.713159    。
        => cheapest way: -0.713159
future costs from 0 to 0 is -1.06726
future costs from 0 to 1 is -1.35359
future costs from 0 to 2 is -1.43306
future costs from 0 to 3 is -2.47373
future costs from 0 to 4 is -2.51762
future costs from 1 to 1 is -0.647842
future costs from 1 to 2 is -0.84418
future costs from 1 to 3 is -1.86847
future costs from 1 to 4 is -1.92873
future costs from 2 to 2 is -0.886027
future costs from 2 to 3 is -1.22063
future costs from 2 to 4 is -1.93378
future costs from 3 to 3 is -1.04067
future costs from 3 to 4 is -1.08455
future costs from 4 to 4 is -0.713159
```

仮説の探索

```
0[] -+- 1[this=>ここ] --- 174[is=>さ れ て] ...
     +- 2[this=>その]
     +- ...
     +- 21[is=>されて]
     +- ....
     +- 162[this is a=>これ は 、] +- 17508[pen .=>ペン で ある 。]
     +- ...
     ...

creating hypothesis 1 from 0
        base score 0
        translation cost -1.35766
        distortion cost 0
        language model cost for ’ここ’ -0.492322
        word penalty 0.361996
        score -1.48798 + futureCost -1.92873 = -3.41672
new best estimate for this stack
merged hypothesis on stack 1, now size 1
creating hypothesis 2 from 0
        base score 0
        translation cost -1.04392
        distortion cost 0
        language model cost for ’その’ -0.522497
        word penalty 0.361996
        score -1.20443 + futureCost -1.92873 = -3.13316
new best estimate for this stack
merged hypothesis on stack 1, now size 2

....

creating hypothesis 162 from 0  <======================
        base score 0
        translation cost -1.03133
        distortion cost 0
        language model cost for ’これ’ -0.437817
        language model cost for ’は’ -0.219777
        language model cost for ’、’ -0.0623787
        word penalty 1.08599
        score -0.665315 + futureCost -1.08455 = -1.74987
new best estimate for this stack
better path, overwriting exisiting hypothesis 156
creating hypothesis 163 from 0
        base score 0
        translation cost -1.85256
        distortion cost 0
```

```
        language model cost for ’これ’ -0.437817
        language model cost for ’は’ -0.219777
        language model cost for ’、’ -0.0623787
        language model cost for ’いわゆる’ -0.876915
        word penalty 1.44798
        score -2.00147 + futureCost -1.08455 = -3.08602
merged hypothesis on stack 3, now size 8

.....

creating hypothesis 174 from 1
        base score -1.48798
        translation cost -2.11081
        distortion cost 0
        language model cost for ’さ’ -1.18326
        language model cost for ’れ’ -0.155323
        language model cost for ’て’ -0.0927082
        word penalty 1.08599
        score -3.9441 + futureCost -1.93378 = -5.87788
merged hypothesis on stack 2, now size 45
creating hypothesis 175 from 1
        base score -1.48798
        translation cost -1.48134
        distortion cost 0
        language model cost for ’に’ -0.46083
        word penalty 0.361996
        score -3.06816 + futureCost -1.93378 = -5.00194
merged hypothesis on stack 2, now size 46
creating hypothesis 176 from 1
        base score -1.48798
        translation cost -1.47625
        distortion cost 0
        language model cost for ’も’ -1.18326
        word penalty 0.361996
        score -3.7855 + futureCost -1.93378 = -5.71929
merged hypothesis on stack 2, now size 47
creating hypothesis 177 from 1
        base score -1.48798
        translation cost -1.29528
        distortion cost 0
        language model cost for ’れる’ -1.18326
        word penalty 0.361996
        score -3.60453 + futureCost -1.93378 = -5.53832
merged hypothesis on stack 2, now size 48

.....
```

```
creating hypothesis 17508 from 162 <============
        base score -0.665315
        translation cost -0.68607
        distortion cost 0
        language model cost for ’ペン’ -1.06335
        language model cost for ’で’ -0.412911
        language model cost for ’ある’ -0.29139
        language model cost for ’。’ -0.0360304
        adding word </s> -5.05187e-07
        word penalty 1.44798
        score -1.70708 + futureCost 0 = -1.70708
new best estimate for this stack
better path, overwriting exisiting hypothesis 17467
creating hypothesis 17509 from 157
        base score -1.33158
        translation cost -1.09842
        distortion cost 0
        language model cost for ’筆’ -1.18326
        word penalty 0.361996
        score -3.25127 + futureCost -0.713159 = -3.96443
better path, overwriting exisiting hypothesis 10164
creating hypothesis 17510 from 157
        base score -1.33158
        translation cost -1.00945
        distortion cost 0
        language model cost for ’ペン’ -1.18326
        language model cost for ’の’ -0.426796
        word penalty 0.723992
        score -3.2271 + futureCost -0.713159 = -3.94026
worse than existing path, discarding

....

best hypothesis now 0
best hypothesis now 2
best hypothesis now 18
 [ 17508 => 162 ]
 [ 162 => 0 ]
BEST: これ は 、 |0.514112|0|2| ペン で ある 。 |0.35283|3|4| -1.70708
```

まとめ

- フレーズベースのSMTのデコーダの基本動作は簡単である
- デコーダの研究は，これからである．

19. 最小誤り率訓練

内山将夫@NICT
mutiyama@nict.go.jp

SMTの構成要素

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \sum_i \lambda_i h_i(\mathbf{e}, \mathbf{f})$$

- 探索： $\arg\max_{\mathbf{e}}$ なる $\hat{\mathbf{e}}$ の探索
- モデリング： 良い素性 $h_i(\mathbf{e}, \mathbf{f})$ の設計
- パラメタ調整：λ_i の学習

最小誤り率訓練 (MERT, Minimum Error Rate Training) は，パラメタ調整に利用される．

パラメタ調整の枠組

- 訓練データ： $h_i(\mathbf{e}, \mathbf{f})$ を獲得する
- 開発データ： λ_i を獲得する
- テストデータ： 翻訳性能を測定する

パラメタ調整の原則

- 翻訳性能を最大化するパラメタが欲しい
  翻訳性能をBLEUで測定するとすると，BLEUを最大化するようなパラメタが欲しい.

開発データにおける入力文を

$$F = \{\mathbf{f}_1, \mathbf{f}_2, ...\}$$

参照用の翻訳文を

$$R = \{\mathbf{r}_1, \mathbf{r}_2, ...\}$$

F を機械翻訳した結果を

$$E = \{\mathbf{e}_1, \mathbf{e}_2, ...\}$$

としたとき,

$$\hat{\lambda} = \arg\max_{\lambda} \mathrm{BLEU}(R, E)$$

なるパラメタ$\hat{\lambda}$が欲しい.

最適化としての$\hat{\lambda}$の探索

1. λ_m = 適当な初期値, $C_i = \phi$
2. for $\mathbf{f}_i \in F$
   (a) $C'_i = \{\mathbf{e}_{i,s} |$ スコア $\Sigma \lambda_m h_m(\mathbf{e}, \mathbf{f}_i)$ が大きい n 翻訳文$\}$
   (b) $C_i = C_i \cup C'_i$. (これまでの翻訳候補に加えて，今のλを利用して得られた翻訳候補を追加する)
3. λを更新する
   (a) 今のλを利用して，拡張されたC_iの中から一番スコアが高い $\mathbf{e}_i = \arg\max_{\mathbf{e}} \Sigma lambda_m h_m(\mathbf{e}, \mathbf{f}_i)$ なる $\mathbf{e}_i$ を得ることにより，$\mathbf{f}_i$ に対する，今のパラメタでの翻訳文とする.
   (b) $E = \{\mathbf{e}_i |$ 上記で選ばれた翻訳文$\}$を利用して，λに対応する $\mathrm{BLEU}(E, R; \lambda)$ を得る.
   (c) これにより，$\lambda \rightarrow \mathrm{BLEU}(E, R; \lambda)$ の関係が計算できるので，λ を少しずつ変えながら，現在の翻訳文集合C_iから，なるべくBLEU が大きくなるように，$\mathbf{e}_i$を選択できるようなλを探す
4. goto 2 or exit

多変量最適化の方法

- Simplex 法，Powell 法等のノンパラメトリック法 (関数勾配が不要な方法)

  cf. Numerical Recipes in C

- 対数線形モデルに特有な方法

対数線形モデルに特有な方法

- ある方向$\mathbf{d}$について，1次元最適化をする
- 上記を，たくさんの方向に繰り返して，少しずつ解を改善して，最適解を求める．

1次元最適化を高速化する．

最小誤り率訓練

BLEU最大化の代りに，より簡単な，誤り個数最小化の問題を考える．これを，あとで，BLEU最大化に拡張する

誤り個数$E(\mathbf{r}_1^s, \mathbf{e}_1^s)$の定義

$$E(\mathbf{r}_1^s, \mathbf{e}_1^s) = \sum_{s=1}^{S} E(\mathbf{r}_s, \mathbf{e}_s)$$

$$\text{参照文のリスト} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_S\}$$

$$\text{翻訳文のリスト} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_S\}$$

$$E(\mathbf{r}_s, \mathbf{e}_s) = \begin{cases} 1 & (\mathbf{r}_s \neq \mathbf{e}_s) \\ 0 & (\mathbf{r}_s = \mathbf{e}_s) \end{cases}$$

この$E(\mathbf{r}_s, \mathbf{e}_s)$が，文が完全に一致するときに$0$で，そうでないときに$1$となっているので，翻訳文の評価としては，ずいぶんと簡略化されている．

誤り個数を最小化するパラメタ $\hat{\lambda}$

$$\hat{\lambda_1^M} = \arg\min_{\lambda_1^M} \sum_{s=1}^{S} E(\mathbf{r}_s, \mathbf{e}(\mathbf{f}_s; \lambda_1^M))$$

$$\mathbf{e}_s = \mathbf{e}(\mathbf{f}_s; \lambda_1^M) = \arg\max_{\mathbf{e} \in C_s} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{f}_s)$$

$$\begin{aligned} C_s &= n\text{個の翻訳候補} \\ \lambda_m &= \text{素性}m\text{の重み} \\ h_m(\mathbf{e}, \mathbf{f}_s) &= \text{素性}m\text{の値} \\ \mathbf{f}_s &= \text{入力文} \end{aligned}$$

誤り個数の性質

$$E(\mathbf{r}_1^s, \mathbf{e}_1^s) = \sum_{s=1}^{S} E(\mathbf{r}_s, \mathbf{e}_s)$$

- $E(\mathbf{r}_s, \mathbf{e}_s)$ の和が全体の値となる
- したがって，個々の誤り $E(\mathbf{r}_s, \mathbf{e}_s)$ と λ_1^M の関係がわかれば，それを加算すれば，全体の誤りと λ_1^M の関係がわかる．

さて，一次元の最適化では，ある方向$\mathbf{d}$に向けての最適化をする．その方向をM次元ベクトル$\mathbf{d}_1^M$により表現する．すると，ある定数ベクトル$\mathbf{g}_1^M$を利用することにより，素性ベクトルλ_1^Mは

$$\lambda_1^M = \mathbf{g}_1^M + \gamma \mathbf{d}_1^M$$

と表現できる．
したがって，λ_1^Mを，ある与えられた方向$\mathbf{d}_1^M$に最適化するとは，$E(\mathbf{r}_1^s, \mathbf{e}_1^s)$が最小となるような，$\mathbf{g}_1^M$と$\gamma$を求めることである．
ここで，

$$\mathbf{e}_s = \mathbf{e}(\mathbf{f}_s; \lambda_1^M) = \arg\max_{\mathbf{e} \in C_s} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{f}_s)$$

により，候補C_sから$\mathbf{e}_s$を選んで，それにより，$E(\mathbf{r}_s, \mathbf{e}_s)$が決まる．この$\mathbf{e}_s$が，$\lambda_1^M$，つまり，$\mathbf{g}_1^M$と$\gamma$により異なる．
したがって，$\mathbf{e}_s$と$\mathbf{g}_1^M$，γの関係が知りたい．

$\mathbf{e}_s$ と $\mathbf{g}_1^M$，γ の関係

素性値のベクトルを $\mathbf{h}_1^M = \{h_1(\mathbf{e}, \mathbf{f}), ...\}$ とする．すると，翻訳文集合 C_s 中の候補を $\mathbf{e}_i$ とすると，そのスコア s_i は，

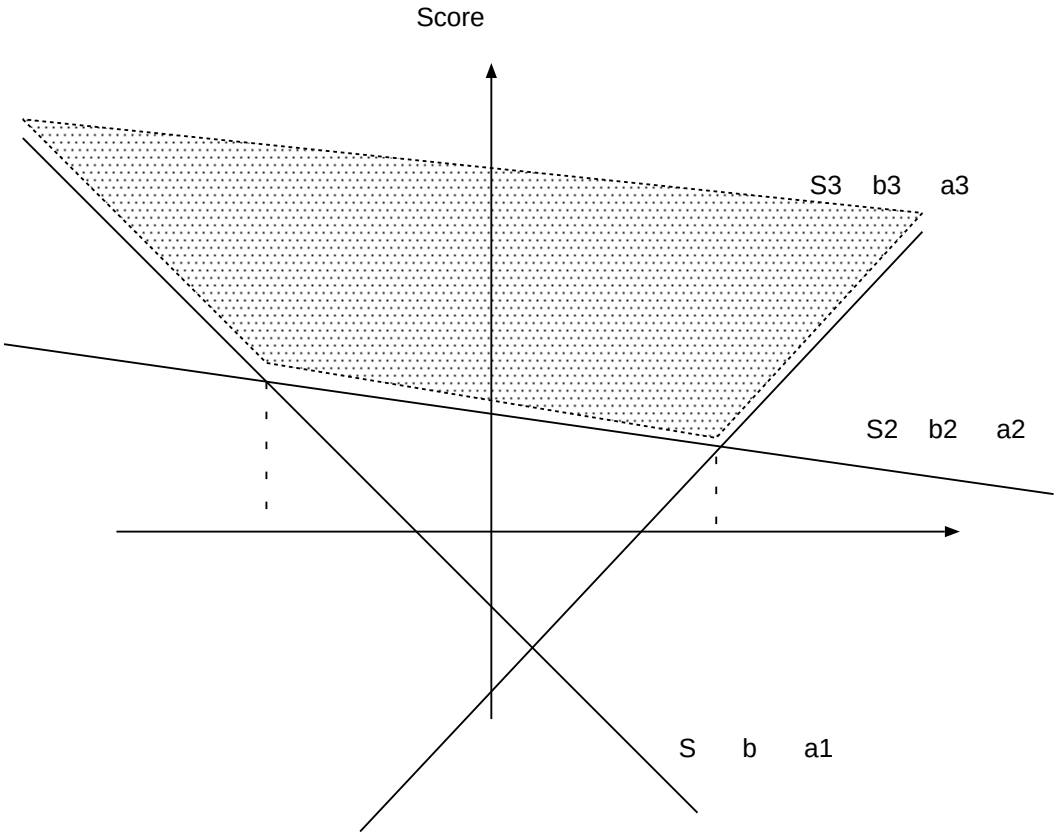
$$\begin{aligned} s_i &= \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}_i, \mathbf{f}_s) \\ &= \lambda_1^M \cdot \mathbf{h}_1^M \\ &= (\mathbf{g}_1^M + \gamma \mathbf{d}_1^M) \cdot \mathbf{h}_1^M \\ &= (\mathbf{g}_1^M \cdot \mathbf{h}_1^M + \gamma \mathbf{d}_1^M \cdot \mathbf{h}_1^M) \\ &= (b_i + \gamma a_i) \end{aligned}$$

ただし，

$$\begin{aligned} b_i &= \mathbf{g}_1^M \cdot \mathbf{h}_1^M \\ a_i &= \mathbf{d}_1^M \cdot \mathbf{h}_1^M \end{aligned} \tag{1}$$

である．つまり，スコア s_i は，ある定数 a_i と b_i から定められる直線 $a_i + \gamma b_i$ の上にある．すなわち，$\mathbf{e}_i$ のスコア s_i は，γ を変えると変わる．

γの変化による$\mathbf{e}_s = \arg\max_{\mathbf{e}_i} s_i$の変化



- $(-\infty, \gamma_1]$ のときは スコアS_1が最大なので文e_1が選ばれる.
- $(-\gamma_1, \gamma_2]$ のときは，e_2が選ばれる
- $(-\gamma_2, \infty]$ のときは，e_3が選ばれる

初期値 $= E(\gamma = -\infty) = E(\mathbf{r}, \mathbf{e}_1)$
左から右に動いていって,
γ_1になったら$\Delta E = E(\mathbf{r}, \mathbf{e}_2) - E(\mathbf{r}, \mathbf{e}_1)$
γ_2になったら$\Delta E = E(\mathbf{r}, \mathbf{e}_3) - E(\mathbf{r}, \mathbf{e}_2)$
のように，γの変化と，その時点でのΔEを記録する.
すると，ある参照文$\mathbf{r}$について，γを動かしていったときに，どの時点で，誤りの個数が変化するかがわかる.

各入力文についての結果を統合する

- 入力文1
  $\gamma_1^1 \rightarrow \Delta E_1^1$
  $\gamma_2^1 \rightarrow \Delta E_2^1$
  ...
- 入力文2
  $\gamma_1^2 \rightarrow \Delta E_1^2$
  $\gamma_2^2 \rightarrow \Delta E_2^2$
  ...

これらをみんなあわせると
$\gamma_1 \rightarrow \Delta E_1$
$\gamma_2 \rightarrow \Delta E_2$
...
のように，どのγにおいて，どの程度，誤りの個数が変化したかがわかる．
これより，$\gamma = -\infty$ のときの誤り個数に対して，$\gamma_1, \gamma_2,$ ... とγを変えていったときの誤りの個数の変化がわかるので，そのときの最小誤りのところのγ を利用する．このγを利用すると，一次元方向での最小化が達成できる．そのため，この結果を利用することにより，多次元の最適化ができる．

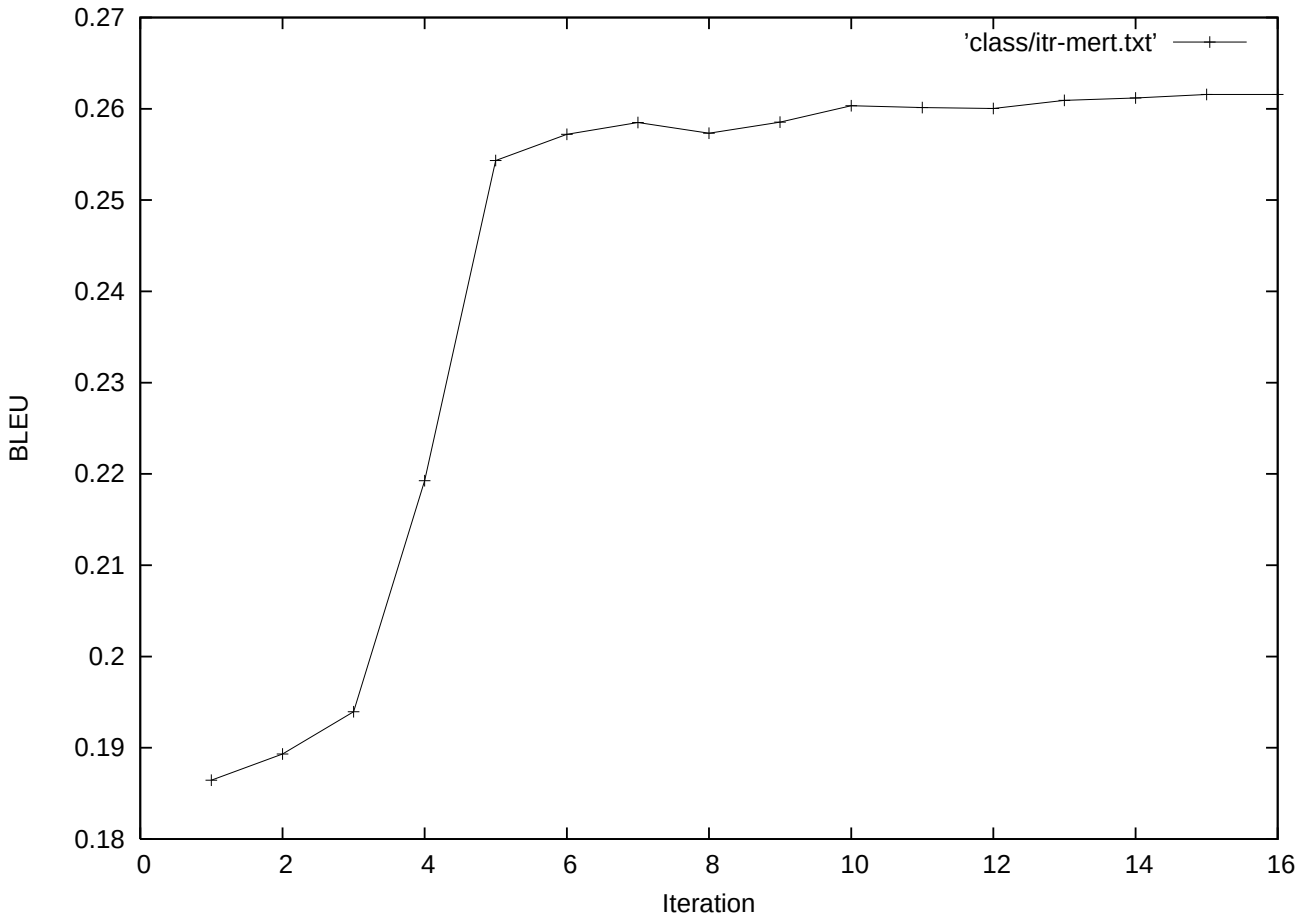
BLEUへの拡張

$$\mathrm{BLEU} = BP(\cdot)\exp(\sum_{n=1}^{N}\frac{\log p_n}{N}) \tag{2}$$

$$\begin{aligned} BP(\cdot) &= \text{長さの短い文へのペナルティ} \\ N &= 4 \\ p_n &= \text{ngram 精度} \\ &= \frac{{}^{\Sigma}\text{MT 訳}\ {}^{\Sigma}\text{MT 訳の ngram}\ \text{共有 ngram 数}}{{}^{\Sigma}\text{MT 訳}\ {}^{\Sigma}\text{MT 訳の ngram}\ \text{ngram 数}} \end{aligned}$$

拡張へのポイントは，BLEU においては，p_n のような部分が，共有する ngram の各文に対する総和として表されることである．したがって，誤りの場合と同様に γ が変化するたびに，共有する ngram の総和が変化するので，そのたびに，共有する ngram の総和等から p_n 等を計算すれば，γ が変化するたびの BLEU の変化がわかる．
したがって，単純な誤り個数の場合と同様に，BLEU が最大となる γ がわかる．

MERTの繰り返し回数とBLEUの関係



BLEUの変化が大きいことがわかる．パラメタ値の調整は，質の良い翻訳を達成するために，必要不可欠である．

繰り返し回数と訳文の変化1

Input: thus , the left input data of node nd15 is obtained .
Reference: これにより、ノードＮＤ１５の左入力データが得られたことになる。

itr1: したがって、左入力データがノードＮＤ１５が得られる。
itr2: 以上のようにして構成されているがされているノードＮＤ１５のようにして得られたままに放置しているとされてい　　るのが、、、、のデータのデータのようにして、入力されているようにされている。
itr3: このようにして、ノードＮＤ１５の左入力データが得られるようになっているようになっている。
itr4: これにより、ノードＮＤ１５の左入力データが得られる。

繰り返し回数と訳文の変化2

Input: the system arrangement shown in fig. 1 will be described in more detail below .
Reference: 図 1 の システム 構成 について 更に 詳細 に 説明 する 。

itr1: この 構成 に 詳しく 説明 する 。
itr2: より 以上 のよう に 構成 されている のは 、 図 1 の 第 1 の 図 に 示 すようにした 場合 について 説明 したように 構成 されている されている システム の 詳細 な 説明 される ようになっている の 下方 に 位置 して 説明 する ように 構成 されている ようにされている 。
itr3: 図 1 に 示 すよう に 構成 されて 、 システム の より 詳細 に 説明 する 以下 のよう に なっ て いる 。
itr4: 図 1 に 示す よう に 構成 さ れ 、 システム の 更に 詳細 に 説明 する 。
itr5: 図 1 に 示す システム 構成 の 詳しく 説明 する 。
itr6: 図 1 に 示す 構成 の システム 詳しく 説明 する 。
itr7: 図 1 に 示す システム 構成 の 更に 詳細 に 説明 する 。

まとめ

- BLEUが最大化するようにパラメタを調整することにより，評価値に沿ったパラメタを獲得できる

これより，適正な評価を与えることが重要であることがわかる．

- 自動的に適正な評価を与えることができれば，その評価を最大化するようにパラメタを調整することにより，良いシステムを作成することができる．

20. まとめ

内山将夫@NICT
mutiyama@nict.go.jp

まとめ

まとめは課題とします

- 講義の内容をまとめたり
- Web 上で利用可能な機械翻訳システムの性能をレポートしたり
- その他，講義と関係のあることをレポートしたり

して下さい．

今後の研究課題

- コーパス
- モデル/アルゴリズム/ソフトウェア
- 評価

コーパス

対訳コーパスが必要だが，大規模な対訳コーパスは少ない
対訳コーパスを収集する効率的な方法が必要である．
既に Web 等にあるものを収集することもできるが，その場合には，著作権を侵害しないようしないといけない．
対訳コーパスがない言語対については，最初から作らないといけない．そのときに，たとえば，標準的な日本語文を何文か用意しておいて，その文に対してさえ，対訳文を用意すれば，どんな言語対についても，コーパスベースの機械翻訳ができるようになれば，すばらしい．

モデル/アルゴリズム/ソフトウェア

モデルは，現在は，何らかの形で構文を利用するものが主流である．けれども，単純なフレーズベースのものと比べて，特に性能が良いことが示されてはいない．単語対応をとるソフトウェアとして GIZA++ が広く使われているが，もうメンテナンスされていないし，性能も最高性能とはいえない．軽く速く動く単語対応をとるソフトウェアが欲しい．

評価

良い翻訳文とは何かが良くわかっていない．
翻訳文の評価は，人手によるものが最も信頼できると思われるが，人手による評価はコストも時間もかかる．
簡単で，かつ，良い自動評価ができれば，機械翻訳の研究は，本質的に発展すると思われる．

結論

コーパスベースの機械翻訳は，まだ始まったばかりであり，これからもっと研究を進める必要がある．