# 翻訳精度の最大化による同時音声翻訳のための文分割法

小田 悠介　Graham Neubig　清水 宏晃　Sakriani Sakti　戸田 智基　中村 哲
奈良先端科学技術大学院大学 情報科学研究科
{oda.yusuke.on9, neubig, hiroaki-sh, ssakti, tomoki, s-nakamura}@is.naist.jp

## 1　はじめに

音声翻訳システムは長年の研究により精度が向上しており，近年様々な分野へと応用の足を伸ばしている．同時性を保ちながら目的言語へと翻訳する同時音声翻訳はその応用分野の一つである．しかし，文を翻訳単位とする従来の音声翻訳 [1] では，文の終了まで翻訳が開始されないため，同時性が大きく損なわれる．このため，機械翻訳の精度を極力維持しつつ入力文を短い単位に分割する文分割法が研究されている [2, 3, 4, 5]．しかし，従来法では分割位置が主にヒューリスティクスに基づいており，翻訳精度に及ぼす影響は直接考慮されていない．また同時性に影響を与える分割単位の平均単語数についても，従来法では明示的な制御ができない．

そこで本研究では，従来法では考慮されていない文分割時の翻訳精度の変化を用いて分割位置を自動的に決定するアルゴリズムを提案する．具体的には，分割後の翻訳精度を最大化する分割位置を貪欲法に基づいて選択する手法を 3 種類定義する．またアルゴリズムのパラメータとして分割単位の平均単語数を導入し，従来法では困難であった平均単語数の明示的な制御を可能とする．

本手法の有効性を検証するために，英日翻訳タスクにおける実験的評価を行った．その結果，従来法と比べて BLEU では同等程度，RIBES では高い精度が得られた．単語数の制御については，指定した単語数からの誤差は 1 単語未満となった．

## 2　同時音声翻訳における文分割

音声翻訳では，入力話者の発話を認識し，翻訳する．対話などの翻訳の場合は発話が比較的短く，発話が終了した時点で翻訳を開始すればよい．しかし講演などでは明らかな発話区切りがない場合が多く，自動的に翻訳を開始するタイミングを判定する必要がある．翻訳の単位として，テキストの翻訳と同じように文を使用するのが自然である [1] が，文の終了までに長い時間を要するため，訳出の同時性が損なわれる．このため，図 1 に示すように文末以外の適当な位置で文を区切って訳出する必要があり，これを行うための文分割法が研究されている．

同時音声翻訳のための文分割法は近年になっていくつか提案されている．Bangalore らは音声認識の無音区間を用いた手法を提案している [2]．Fujita らは単語アラインメントがその位置で交差するかどうかの確率（右確率）を用いて分割を行う手法を提案しており，無音区間による手法と比べて精度を維持したまま訳出速度の向上を実現している [3]．Rangarajan らによる

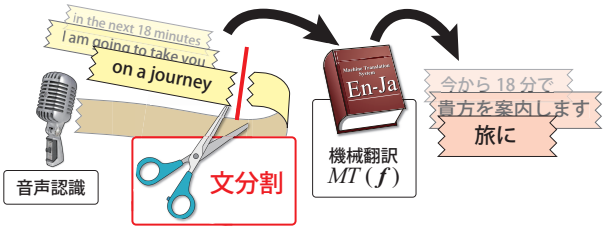


図 1: 同時音声翻訳における文分割の位置付け

報告では英西翻訳タスクに対して複数の手法が評価されており，コンマ，ピリオドの位置を SVM で予測する手法と接続詞で分割する手法が最も高い精度であると述べられている [4]．清水らによる研究では，英日翻訳の同時通訳者の訳出タイミングを分析し，英語の品詞が特定の組み合わせで現れる場合に分割位置とする手法を提案しており，コンマ，ピリオドの予測による手法と同等程度の精度を実現している [5]．

ここに挙げたいずれの手法も音韻的，言語的な特徴のみを使用して分割位置を決定しており，分割位置が機械翻訳の精度にどのような影響を与えるかは直接考慮されていない．また，文分割で生成される単語列の平均単語数は訳出時間に影響を与えるが，同じくここに挙げたいずれの手法も，平均単語数を直接制御できる手法ではない[1]．次節以降では，分割位置による翻訳精度の変化，及び分割単位の平均単語数を指標に用いる文分割法を提案する．

## 3　翻訳精度の最大化による文分割

提案法では学習済みの機械翻訳システムの出力を用いて文分割モデルの学習を行う．以下，学習に用いる対訳データの原言語側の文を $\mathcal{F} = \{\boldsymbol{f}_j | 1 \leq j \leq N\}$，目的言語側の参照文を $\mathcal{E} = \{\boldsymbol{e}_j | 1 \leq j \leq N\}$ と表す．$N$ は対訳文の数である．本来の音声翻訳では文末が明示されないため推定する必要があるが，ここでは文末推定は事前に行われているものとする．また，機械翻訳システムを原言語文 $\boldsymbol{f}$ の関数 $MT(\boldsymbol{f})$ として表す．

### 3.1　アルゴリズムの概要

次節以降で複数の分割アルゴリズムを提案するが，いずれの手法も以下に示す基本的な手順に従う．

1. アルゴリズムのパラメータとして，分割単位の平均単語数 $\mu$ と機械翻訳の評価尺度 $EV(\boldsymbol{e}_{hyp}, \boldsymbol{e}_{ref})$ を決める．$EV$ としては BLEU [6] などの自動評

[1] 右確率による手法は分割の頻度を決めるパラメータを持つが，このパラメータから平均単語数を推定することは難しい．

価尺度を選択できる．$\mu$ から $\mathcal{F}$ 全体の分割数 $K$ を式 (1) で求める．

$$K := \max\left(0, \left\lfloor \frac{\sum_{\boldsymbol{f}\in\mathcal{F}} |\boldsymbol{f}|}{\mu} \right\rfloor - N\right) \quad (1)$$

2. $\mathcal{F}$ 中の全ての分割可能な位置から $K$ 個を選択し，分割位置の集合 $\mathcal{S}^*$ とする．このとき $K$ 個の分割位置は，式 (2) に示すように，ある評価関数 $\omega$ をなるべく大きくするものを選択する．

$$\mathcal{S}^* \simeq \underset{\mathcal{S}\in\{\mathcal{S} \mid |\mathcal{S}|=K\}}{\arg\max} \omega(\mathcal{S}|\mathcal{F}, \mathcal{E}, EV, MT) \quad (2)$$

以下，簡単のために $\omega$ の条件を省略する．本研究では基本的に，$\omega$ として式 (3) に示す対訳文ごとの評価尺度の総和を用いる．

$$\omega(\mathcal{S}) := \sum_{n=1}^{N} EV(MT(\boldsymbol{f}_n|\mathcal{S}), \boldsymbol{e}_n) \quad (3)$$

ここで，$MT(\boldsymbol{f}|\mathcal{S})$ は原言語文 $\boldsymbol{f}$ を分割位置集合 $\mathcal{S}$ で分割し，それぞれの分割単位に対して機械翻訳を行い，結果を順に結合したものである．式の制約から，機械翻訳の評価尺度 $EV$ は各対訳文ごとに独立して計算できる必要がある．

3. $\mathcal{S}^*$ を適当な学習器によって学習し，未知のデータに対する予測を行うモデル $M^*$ を学習する．本研究では線型 SVM [7] を用いて学習した．素性には分割候補となる位置の前後 2 単語による単語 1, 2, 3-gram，及び品詞 1, 2, 3-gram を用いた．

異なる分割位置集合同士の $\omega$ の関係は不明であるため，最適な分割位置集合 $\mathcal{S}^*$ を厳密に求めるには，可能な分割位置集合全てに対して総当たりで $\omega$ を評価しなければならない．しかし，1 文あたりの可能な分割位置の組み合わせは $2^{|\boldsymbol{f}|-1}$ 通り存在し，$\mathcal{F}$ 全体での組み合わせの数はこの総乗となるため，全ての仮説を調べるのは現実的に不可能である．このため，$\omega$ に関して何らかの仮定を置くことで近似解法を導入する必要がある．以下では 3 種類の手法を提案する．

## 3.2 貪欲法による分割位置の選択

最初の近似解法として，分割位置を一つずつ貪欲 (Greedy) 的に決めていく手法を述べる．この手法では，$k$ 番目の分割位置を決める際に $k-1$ 番目までに選択された分割位置は変化させず，まだ分割されていない位置から式 (3) の値が最大になるものを一つだけ選んで追加する．この操作を分割位置の個数が $K$ に達するまで繰り返す．単一の文に対する例を図 2 に示す．

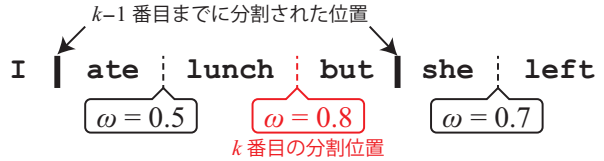


図 2: 貪欲法による分割位置の選択

Algorithm 1 に，貪欲法により学習データの分割位置を求めるアルゴリズムを示す．

**Algorithm 1** Greedy Segmentation Search

```
S* ← ∅
for k = 1 to K do
  ω* ← −∞, s* ← nil
  for all s ∈ {s|s ⊄ S*} do
    ω' ← ω(S* ∪ {s})
    if ω' > ω* then
      ω* ← ω', s* ← s
    end if
  end for
  S* ← S* ∪ {s*}
end for
return S*
```

## 3.3 素性による分割位置のグループ化

前節で述べた手法は学習データに対して高い翻訳精度を実現する分割位置を探し出すことができると考えられる．しかし，翻訳システム $MT$ と評価尺度 $EV$ は両方とも複雑であり，この複雑さにより評価関数 $\omega$ には一定のノイズが存在する．このため純粋に $\omega$ の値のみを用いる貪欲法では，学習データ中で機械翻訳の結果が偶然高い評価値となった分割位置を多く選び出す可能性がある．このことは，学習データに対する文分割の精度は上がるが，他のデータに対して学習結果を適用した場合に精度を下げてしまうことにつながる．

分類器の学習データとして利用する分割位置を選択する際，より一貫性のあるものが選ばれるようにすれば，この問題に対処できると考えられる．本研究では，特定の位置で分割する場合，その位置と類似する特徴を持つ他の位置も同様に分割するという制約を設けることで，この問題を解決する手法を提案する．具体的には図 3 に示す例のように，原言語文に含まれる全ての分割可能な位置を素性でグループ化し，同じグループに属する位置を必ず同時に分割することにする．

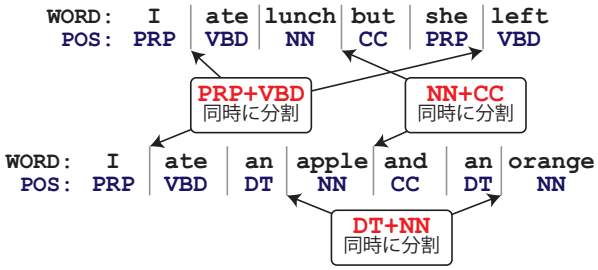


図 3: 素性による分割位置のグループ化

この制約を設けることによって，特定の位置での分割で良い評価値が得られるような場合でも，同じ素性を持つ他の位置で悪い評価値を得るようなものは分割位置として選択されにくくなることが期待できる．また，同じ素性を持つ位置は全て分割されるか否かのどちらか一方となるため，学習結果の素性集合に含まれるかどうかだけを調べれば分割位置を決定できる．このため，アルゴリズムの結果を更に別の学習器を用いて学習する必要がなくなるという利点もある．

この手法では一度に分割される位置が複数存在するため，アルゴリズムを変更する必要が生じる．Algorithm 2 に示すのは，今まで調べた分割数に対する結果を記憶しておき，動的計画法 (Dynamic Program-

ming: DP) により現在の分割数に対する最善の分割位置の素性集合を求めるアルゴリズムである．ここで，$c(\phi|\mathcal{F})$ は素性 $\phi$ が原言語文の集合 $\mathcal{F}$ に現れる回数，$\mathcal{S}(\Phi)$ は素性集合 $\Phi$ により決まる分割位置集合を表す．

**Algorithm 2** Greedy+DP Segmentation Search

$\Phi_0 \leftarrow \varnothing$
**for** $k = 1$ **to** $K$ **do**
  $\omega^* \leftarrow -\infty$, $\Phi_k \leftarrow nil$
  **for** $j = 0$ **to** $k-1$ **do**
    **for all** $\phi \in \{\phi | c(\phi|\mathcal{F}) = k - j \wedge \phi \notin \Phi_j\}$ **do**
      $\omega' \leftarrow \omega(\mathcal{S}(\Phi_j \cup \{\phi\}))$
      **if** $\omega' > \omega^*$ **then**
        $\omega^* \leftarrow \omega'$, $\Phi_k \leftarrow \Phi_j \cup \{\phi\}$
      **end if**
    **end for**
  **end for**
**end for**
**return** $\Phi_K$

本稿の実験では，分割位置の前後 2 単語による品詞 2-gram を素性として使用した．これは，同時通訳者の訳出タイミングを分析した清水らの結果 [5] に基づく．

### 3.4 素性の数による正則化

素性による分割位置のグループ化という制約を設けた場合でも低頻度の素性に対してはノイズが残り，これらが多数選択されてしまう可能性がある．この問題を避けるために，選択した素性の数に対する正則化係数 $\alpha$ を導入し，分割位置に関する素性集合 $\Phi$ を選択する際の評価関数 $\omega$ を式 (4) のように再定義する．

$$\omega(\Phi) := \sum_{n=1}^{N} EV(MT(\boldsymbol{f}_n|\mathcal{S}(\Phi)), \boldsymbol{e}_n) - \alpha|\Phi| \quad (4)$$

$\alpha$ として大きな値を選ぶと，新たな素性を追加することによるペナルティが大きくなり，結果として学習データに頻繁に現れる素性を重視するようなモデルを学習することになる．逆に $\alpha$ に小さな値を選ぶと，出現頻度の低い素性を多く取り入れるようなモデルを学習するようになる．$\alpha = 0$ のときは素性によるグループ化のみを考慮する場合と一致する．

## 4 実験的評価

### 4.1 実験設定

以上で述べた 3 種類の手法について有効性を検証するために，機械翻訳のタスクを用いて実験した．用いたデータは TED 講演の英日翻訳結果，英辞郎辞書 (EIJI) 及び例文 (REIJI) である．表 1 に使用したデータの詳細を示す．

表 1: 実験に用いたデータの詳細

| データ | データセット | 単語 (en) | 単語 (ja) |
|---|---|---|---|
| PBMT 学習 | TED,EIJI,REIJI | 13.7M | 19.7M |
| 文分割学習 | TED | 159k | 215k |
| テスト | TED | 8.21k | 11.9k |

英語の単語分割と品詞推定に Stanford POS Tagger [8]，日本語の単語分割に KyTea [9] を使用した．機械翻訳システムには Moses [10] により学習された PBMT システムを使用した．単語の並べ替え制限は精度が最大となった 12 単語とし，その他の設定についてはデフォルトとした．提案法の学習に用いる評価尺度として BLEU を選択した．ただし，文ごとに BLEU を算出すると多くの文で値が 0 となってしまうため，実際には内部計算で用いるスコアを修正した BLEU+1 [11] を尺度として用いた．テストデータの翻訳結果に関しては BLEU と，より語順に厳しい評価尺度である RIBES [12] を用いて評価した．提案法は順に単純な貪欲法を Greedy+SVM，素性で分割位置をグループ化する手法を Greedy+DP，グループ化と正則化を加えた手法を Greedy+DP+$\alpha$ と表記する．Greedy+DP+$\alpha$ における式 (4) の正則化係数 $\alpha$ は 0.1 と 0.5 の 2 種類について調べた．提案法との比較対象として，コンマ，ピリオド位置の予測による手法 (Punct-Predict)，右確率による手法 (RP)，及びランダムに分割位置を選択した場合 (Random) について併記した．

### 4.2 実験結果

図 4，5 に，各手法によるテストデータの翻訳結果の BLEU と RIBES をそれぞれ示す．

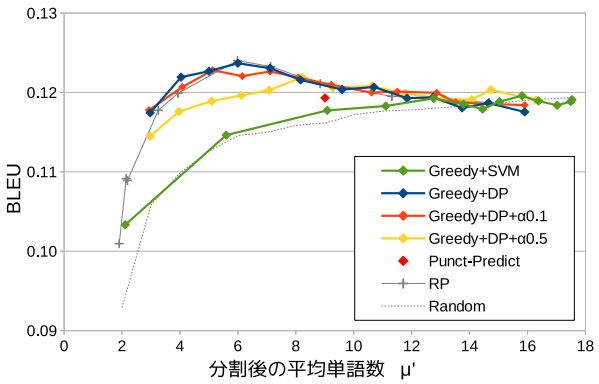


図 4: テストデータの翻訳精度 (BLEU)

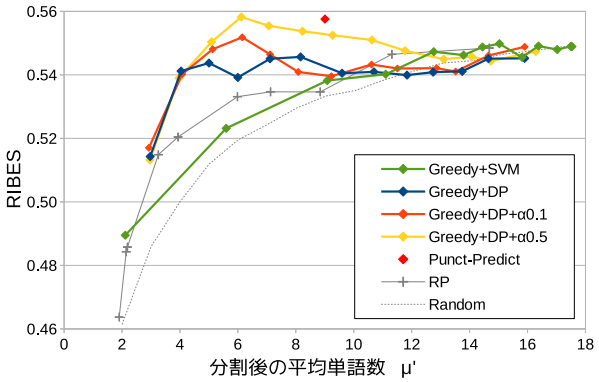


図 5: テストデータの翻訳精度 (RIBES)

単純な貪欲法による手法ではいずれの評価尺度も従来法に比べて低い精度を示しており，ランダム選択による結果に近い値となっている．参考のために，各アルゴリズムで学習データ自身を分割した際の BLEU を

図 6 に示す．この図を見て分かるように，単純な貪欲法では学習データに対して非常に高い精度を示している．これらの結果から，単純な貪欲法では学習データで偶然翻訳精度が高くなる分割位置が多く選択され，汎用的なモデルを学習できなかったことが推測できる．

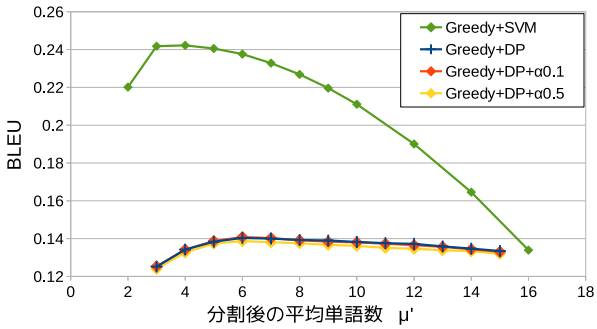


図 6: 学習データの翻訳精度 (BLEU)

一方，素性でグループ化する手法では BLEU において右確率による手法と同等の精度を示しており，RIBES では分割単位の平均単語数が 8 単語以下の領域で右確率よりも良い精度を示している．グループ化と正則化を加えた手法では正則化係数を大きくしたときに若干の BLEU の低下が見られたが，RIBES は逆に大きく上昇するという結果となった．コンマ，ピリオドの予測による手法と比較した場合，$\alpha = 0.5$ の場合において，より小さな平均単語数で同等程度の精度を実現していることが分かる．ただし，BLEU を用いて最適化したにも関わらず BLEU が低下し，RIBES が上昇したことについての原因は現時点では不明であり，今後の検証が必要である．

素性によるグループ化を行う 2 手法について Bootstrap Resampling による有意水準 5%の検定 [13] を行った結果，BLEU に関しては右確率による手法と特に有意な差は認められなかった．RIBES に関しては平均単語数が 8 単語以下のほぼ全ての設定において，右確率に対して有意に良い性能を示した．語順の正確さを重視する RIBES でより高い評価値が得られたことは，本手法が英語と日本語のような語順の差が大きい言語対に対してより効果的であることを示唆している．

次に，図 7 に各アルゴリズムについて学習時に指定した平均単語数 $\mu$ と，学習結果を用いてテストデータを分割した際の平均単語数 $\mu'$ の絶対誤差を示す．

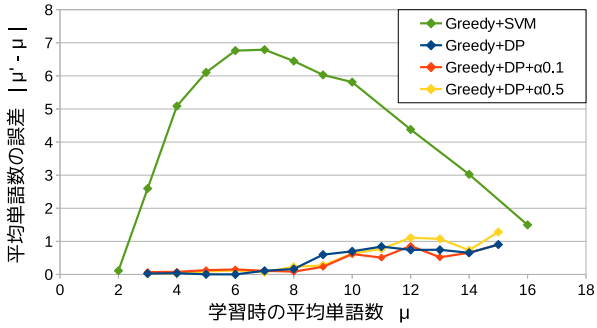


図 7: 平均単語数の絶対誤差

素性によるグループ化を行う 2 手法では，いずれの設定においても平均単語数の誤差をほぼ 1 単語以内に収めることに成功している．このことは，これらの手法が分割単位の平均単語数を学習時のパラメータとして明示的に制御できることを示唆している．

## 5 おわりに

同時通訳システムに用いる文分割アルゴリズムとして，学習データに対する評価尺度の最大化を基準とするアルゴリズムを提案し，従来法と性能を比較した．その結果，BLEU による評価では従来法と同等程度，RIBES による評価では従来法よりも高い精度が得られた．また本手法では，分割単位の平均単語数をほぼ 1 単語以下の誤差で制御できることが分かった．

今後の課題としては，これらのアルゴリズムの高速化や精度面での改良，他の言語対に対する本手法の評価などが挙げられる．

## 謝辞

本研究の一部は JSPS 科研費 24240032 の助成を受け実施したものである．

## 参考文献

[1] Evgeny Matusov, Arne Mauser, and Hermann Ney. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proc. IWSLT*, pages 158–165, 2006.

[2] Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez. Real-time incremental speech-to-speech translation of dialogs. In *Proc. NAACL HLT*, pages 437–445, 2012.

[3] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *InterSpeech*, 2013.

[4] Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan. Segmentation strategies for streaming speech translation. In *Proc. NAACL HLT*, pages 230–238, 2013.

[5] 清水 宏晃, Graham Neubig, Sakriani Sakti, 戸田 智基, and 中村 哲. 同時通訳データを利用した同時音声翻訳のための訳出タイミング決定手法. In **言語処理学会第** *20* **回年次大会** *(NLP2014)*, 2014.

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

[7] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, pages 1871–1874, 2008.

[8] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. NAACL*, pages 173–180, 2003.

[9] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proc. NAACL HLT*, pages 529–533, 2011.

[10] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, pages 177–180, 2007.

[11] Chin-Yew Lin and Franz Josef Och. Orange: A method for evaluating automatic evaluation metrics for machine translation. In *Proc. COLING*, 2004.

[12] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proc. EMNLP*, pages 944–952, 2010.

[13] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*, pages 388–395, 2004.