ALAGIN 機械翻訳セミナー -
同時音声翻訳＋音響情報の翻訳

Graham Neubig
奈良先端科学技術大学院大学（ NAIST ）
2014/3/7

音声翻訳の今

音声翻訳の始まり

音声翻訳システム

音声認識

こんにちは、駅はどこですか？

機械翻訳

Hello, where is the station?

音声合成

同時性の高い音声翻訳

実際の通訳

音声翻訳システム

- ある言語の音声から違う言語の音声へ翻訳

音声認識

こんにちは、駅はどこですか？

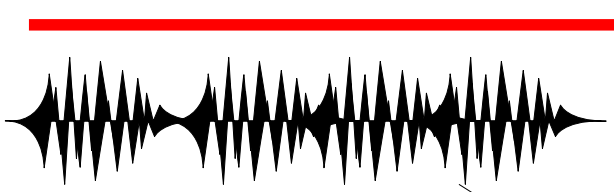
機械翻訳

Hello, where is the station?

音声合成

遅延の問題

- 従来のシステムは１文の入力が終わるまで翻訳しない！



遅延

音声認識

こんにちは、駅はどこですか？

機械翻訳

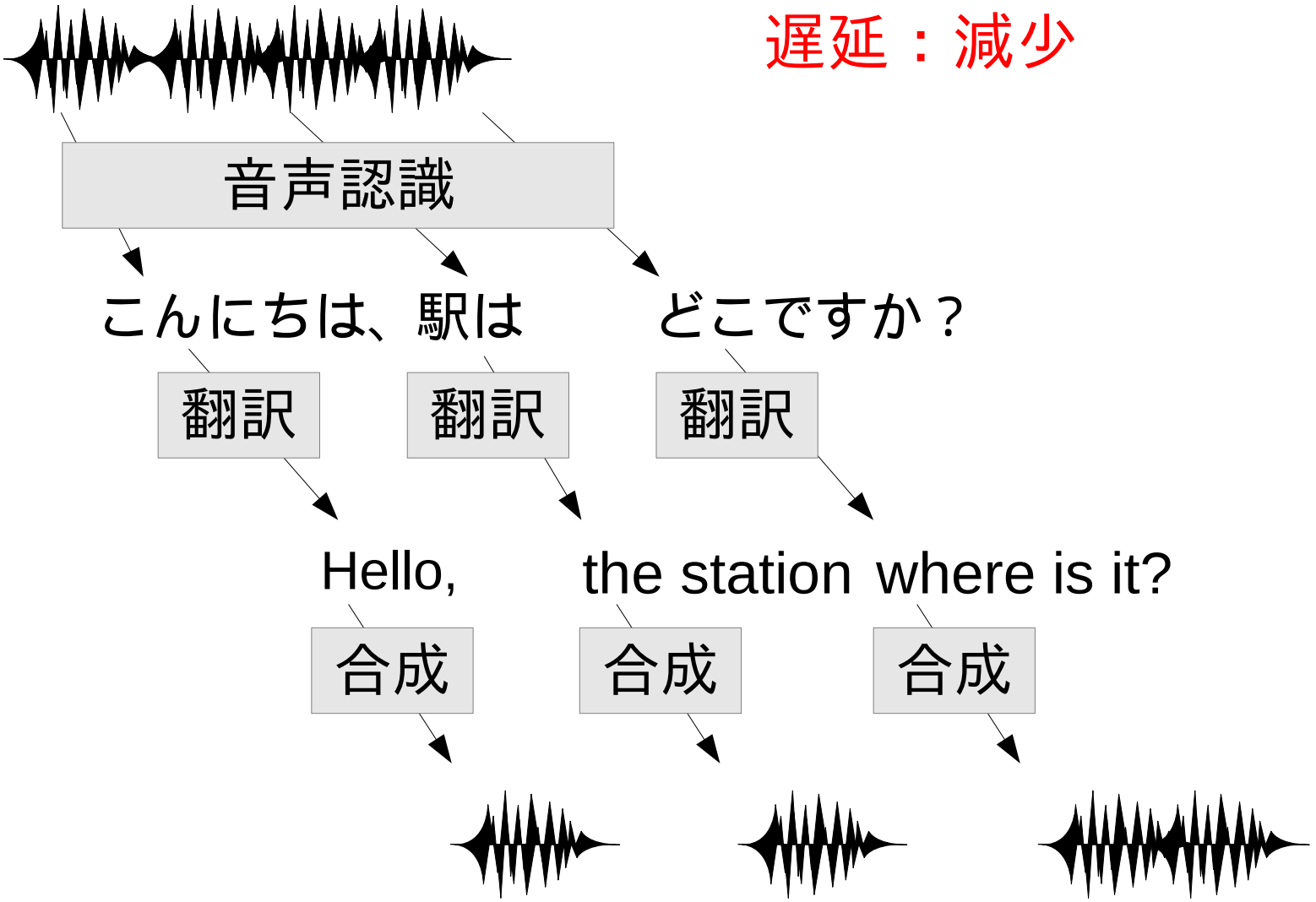
Hello, where is the station?

音声合成

目標：遅延の低減

- １文が完全に終わる前に適切なタイミングで翻訳開始

遅延：減少



遅延の削減に関わる研究

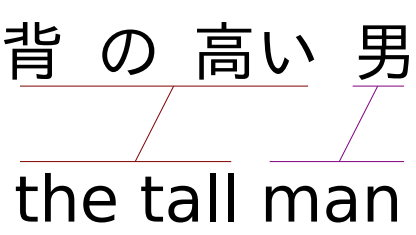
- [Fugen+ 2007]
  - 言語モデルや音響情報に基づくタイミング決定
- [Bangalore+ 2012]
  - 音声認識の無音区間（pausing）を用いて文を分割
- [Rangarajan+ 2013]
  - 予測された句読点の挿入位置（コンマ、ピリオド、その他）を使用
  - 線型 SVM で学習（素性：word 1,2,3-gram / POS 1, 2, 3-gram)
  - 数種類の手法を比較検討　…　句読点による手法が最高性能

並び替え確率に基づく訳出タイミング決定 [Fujita+ 13]

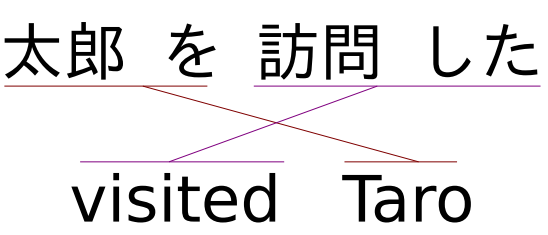
並べ替えモデル

- 単語の並べ替え方を確率的に表し、精度向上に貢献
- 現在の単語と次の単語の順番は４種類に分類：

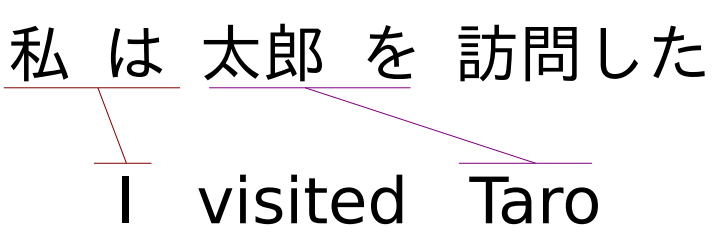
順：順番は同じ



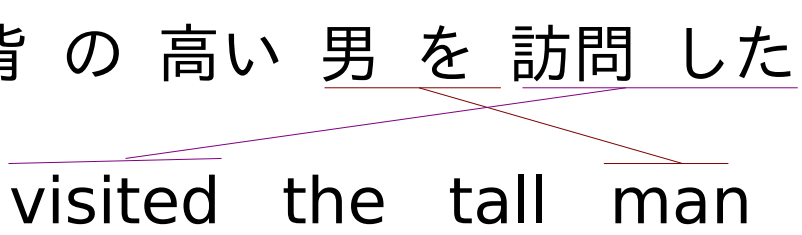
逆順：順番は逆



不連続（右）：



不連続（左）：



- 「順」と「不連続(右)」の確率の和は「右確率」

手法１：
フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

手法１：
フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

hello　　where　　is　　the　　station

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

hello　　where　　is　　the　　station

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

hello　　where　　is　　the　　station

“hello”
モデルに存在
↓
保留

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

hello　　where　　is　　the　　station

“hello”
モデルに存在
↓
保留

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

~~hello~~　　where　　is　　the　　station

“hello”
モデルに存在
↓
保留

“hello where”
存在しない
↓
出力
“hello”

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

~~hello~~　　where　　is　　the　　station

“hello”
モデルに存在
↓
保留

“hello where”
存在しない
↓
出力
“hello”

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

~~hello~~　　where　　is　　the　　station

“hello”
モデルに存在
↓
保留

“hello where”
存在しない
↓
出力
“hello”

“where is”
モデルに存在
↓
保留

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

~~hello~~　　where　　is　　the　　station

“hello”
モデルに存在
↓
保留

“hello where”
存在しない
↓
出力
“hello”

“where is”
モデルに存在
↓
保留

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

~~hello~~　　~~where is~~　　the　　station

“hello”	“hello where”	“where is”	“where is the”
モデルに存在	存在しない	モデルに存在	存在しない
↓	↓	↓	↓
保留	出力 “hello”	保留	出力 “where is”

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

~~hello~~　　~~where is~~　　the　　station

“hello”	“hello where”	“where is”	“where is the”
モデルに存在	存在しない	モデルに存在	存在しない
↓	↓	↓	↓
保留	出力 “hello”	保留	出力 “where is”

手法１：フレーズを用いた訳出タイミング決定

- 認識された単語を１語ずつ入力
- 単語列が翻訳モデルに存在する限り翻訳しない

翻訳モデル

hello→ こんにちは　　where is→ どこですか　　the station→ 駅
where→ どこ　　the→ その

入力文字列

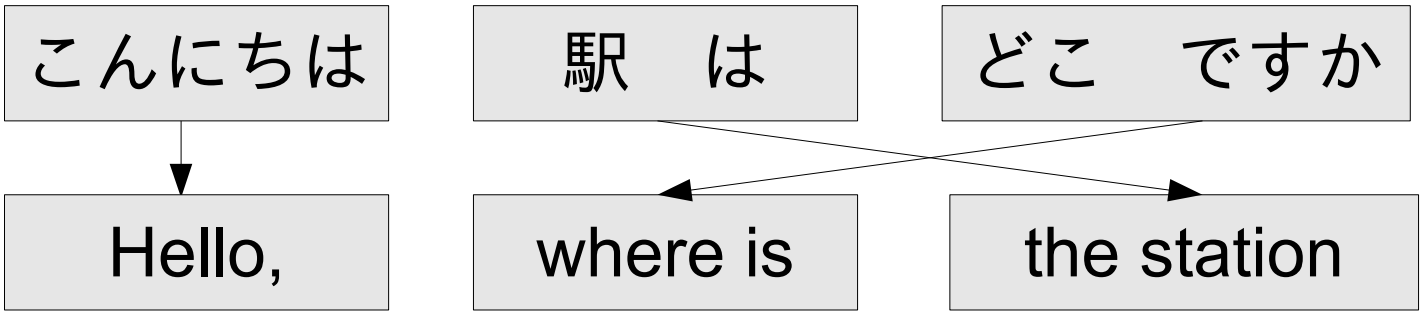
~~hello~~　　~~where is~~　　~~the station~~

“hello”	“hello where”	“where is”	“where is the”	“the station”
モデルに存在	存在しない	モデルに存在	存在しない	発話終了
↓	↓	↓	↓	↓
保留	出力 “hello”	保留	出力 “where is”	出力 “the station”

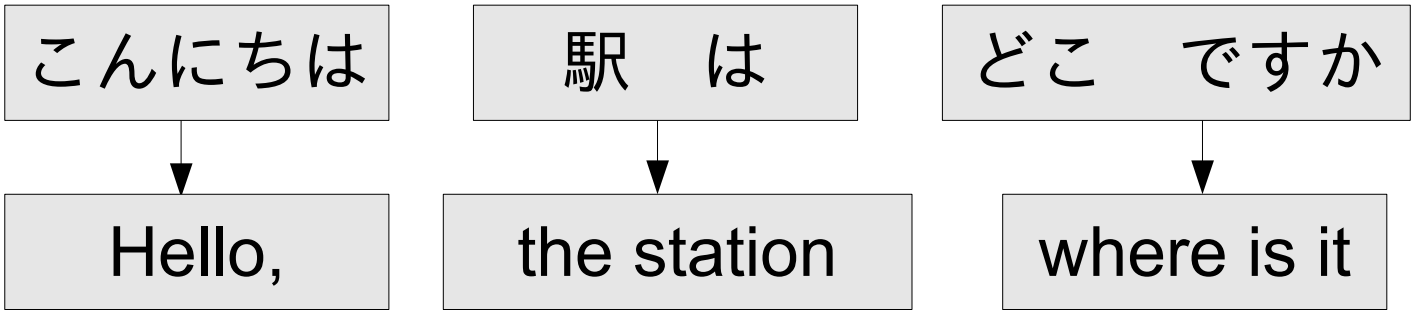
手法１の問題点

- 翻訳精度の低下につながる場合も

通常のフレーズベース翻訳



手法１を用いた場合

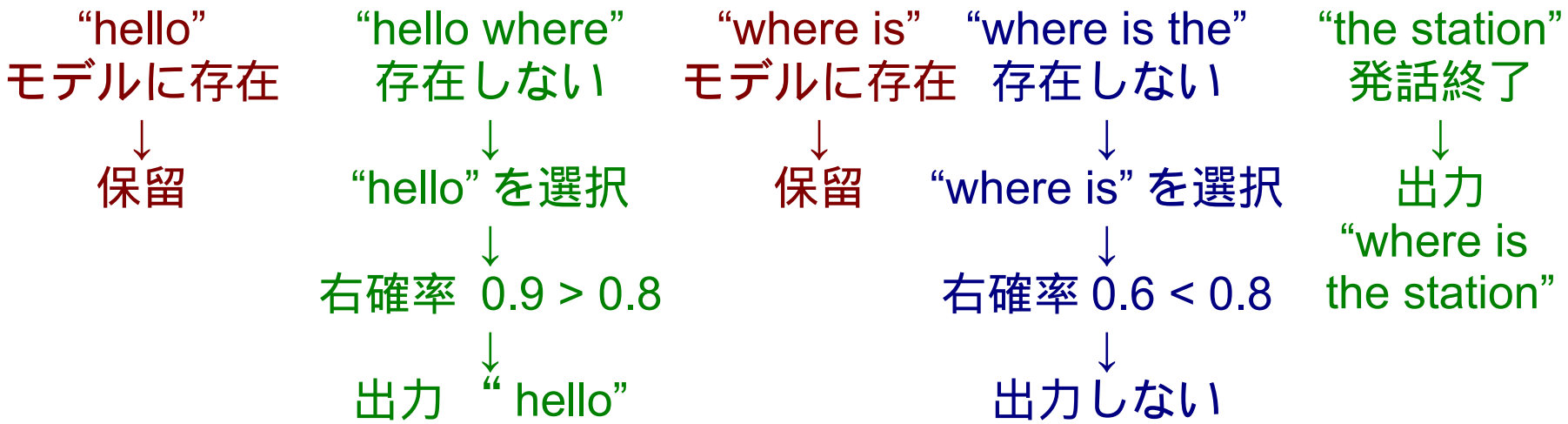


手法２：右確率を用いた訳出タイミングの調整

- まず、手法１を用いて訳出タイミングを仮確定
- フレーズの右確率が閾値を上回った場合のみ本確定

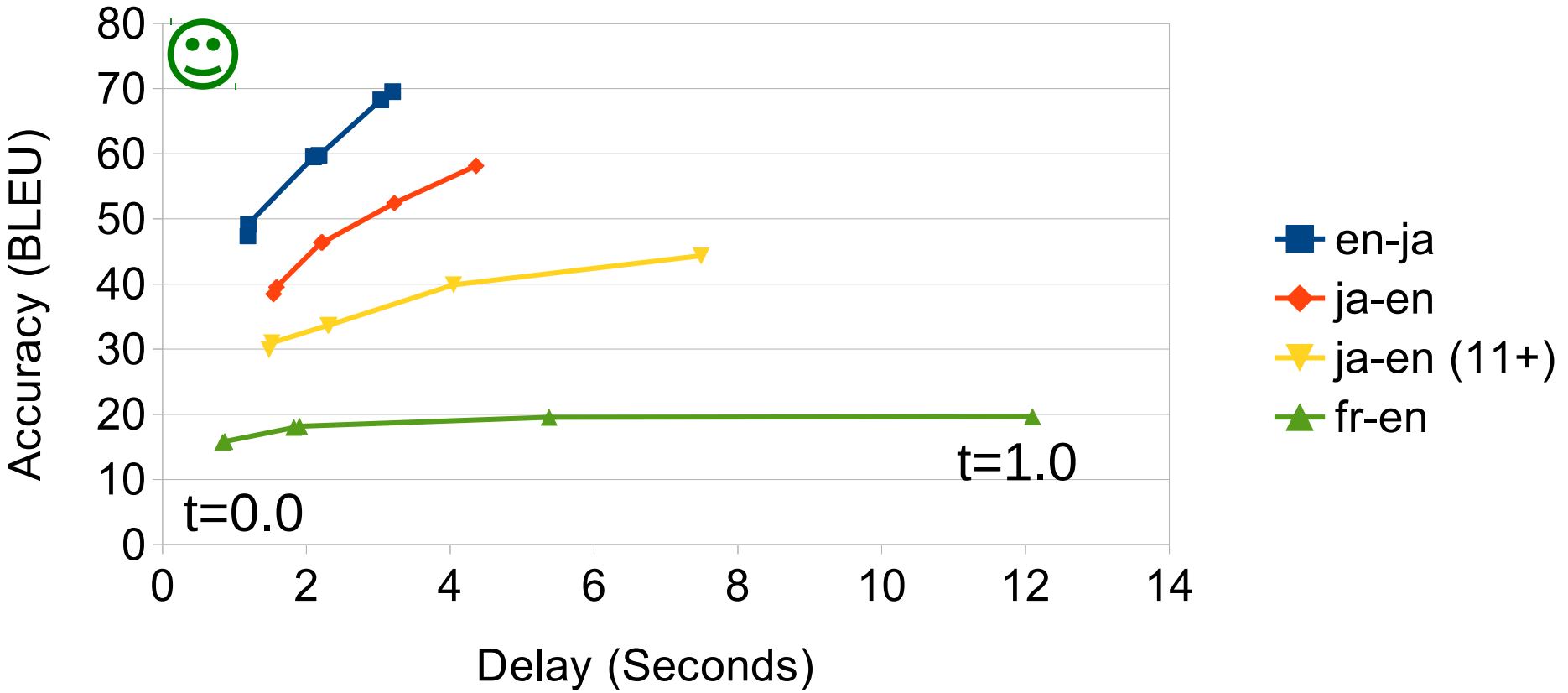
例 (閾値 = 0.8):

hello　　where　　is　　the　　station



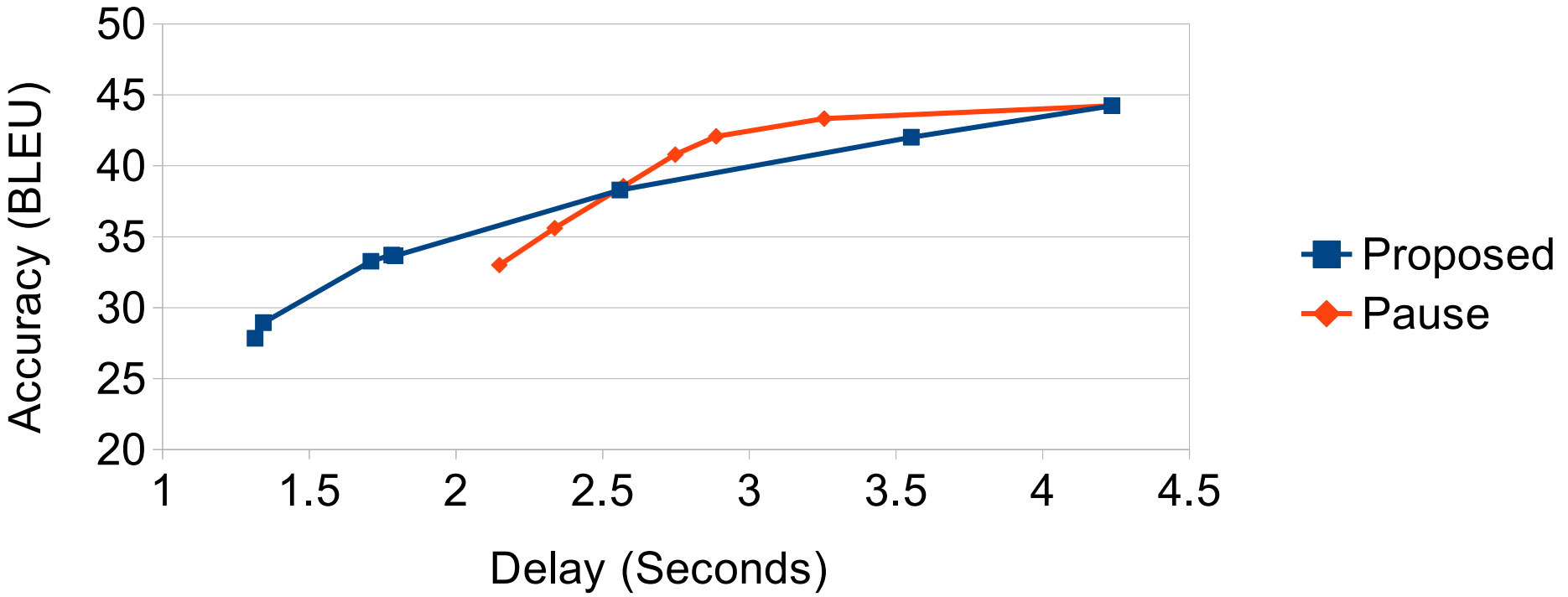
- 閾値が 1.0 の場合は文ごと、 0.0 の場合はフレーズごと

提案手法の精度・遅延



- 全ての設定において遅延が減少
- 長い文、語順が類似している言語対で特に顕著

実験結果 2: ポーズを用いたタイミング決定との比較



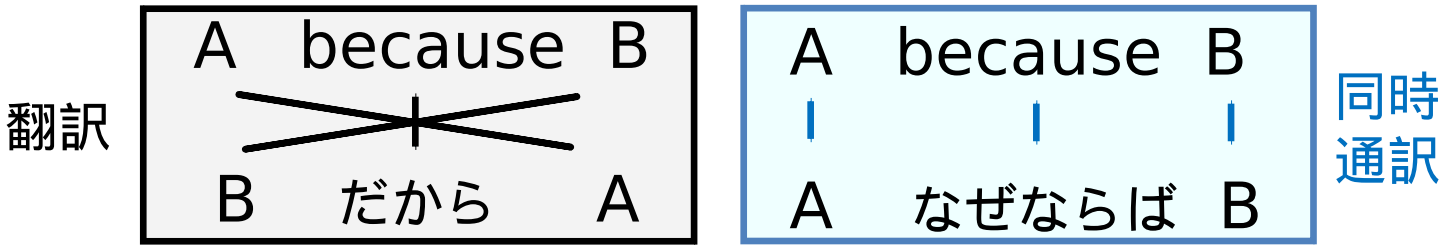
- 速い訳出では提案手法、遅い訳出ではポーズを用いた分割が有効

システムデモ

同時通訳データを用いた同時音声翻訳
[Shimizu+ 13, Shimizu+ 14]

同時通訳者の技術

- 語彙の選択
  - 文法構造が異なる言語対において並び替えを減少させる狙い



- サラミテクニック [Jones 02]
  - １つの長い文を複数のチャンクに分割

last year ↘ 去年　　I went to Japan ↘ 日本に行った

遅延時間を短縮するために
同時通訳者は様々な技術を駆使

31

同時通訳データ

- 収録材料
  - TED 講演（英語 → 日本語）

利点：翻訳データ（字幕）と同時通訳データを比較

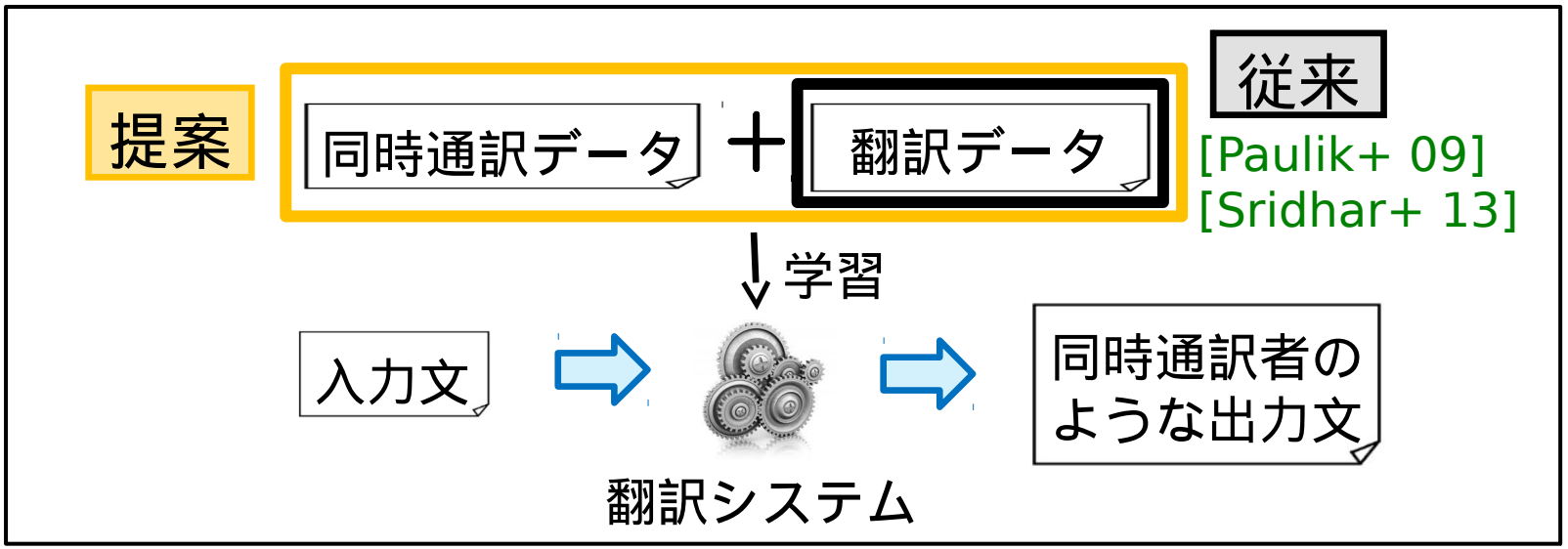
- 同時通訳者
  - 通訳経験年数が異なる三人
  - 経験が長い順に S, A, B ランク

Experience	Rank
15 years	S rank
4 years	A rank
1 year	B rank

利点：同時通訳の訳出の違いを分析

同時通訳データの適用

- アプローチ
  - 同時通訳者のように訳出する同時音声翻訳の構築

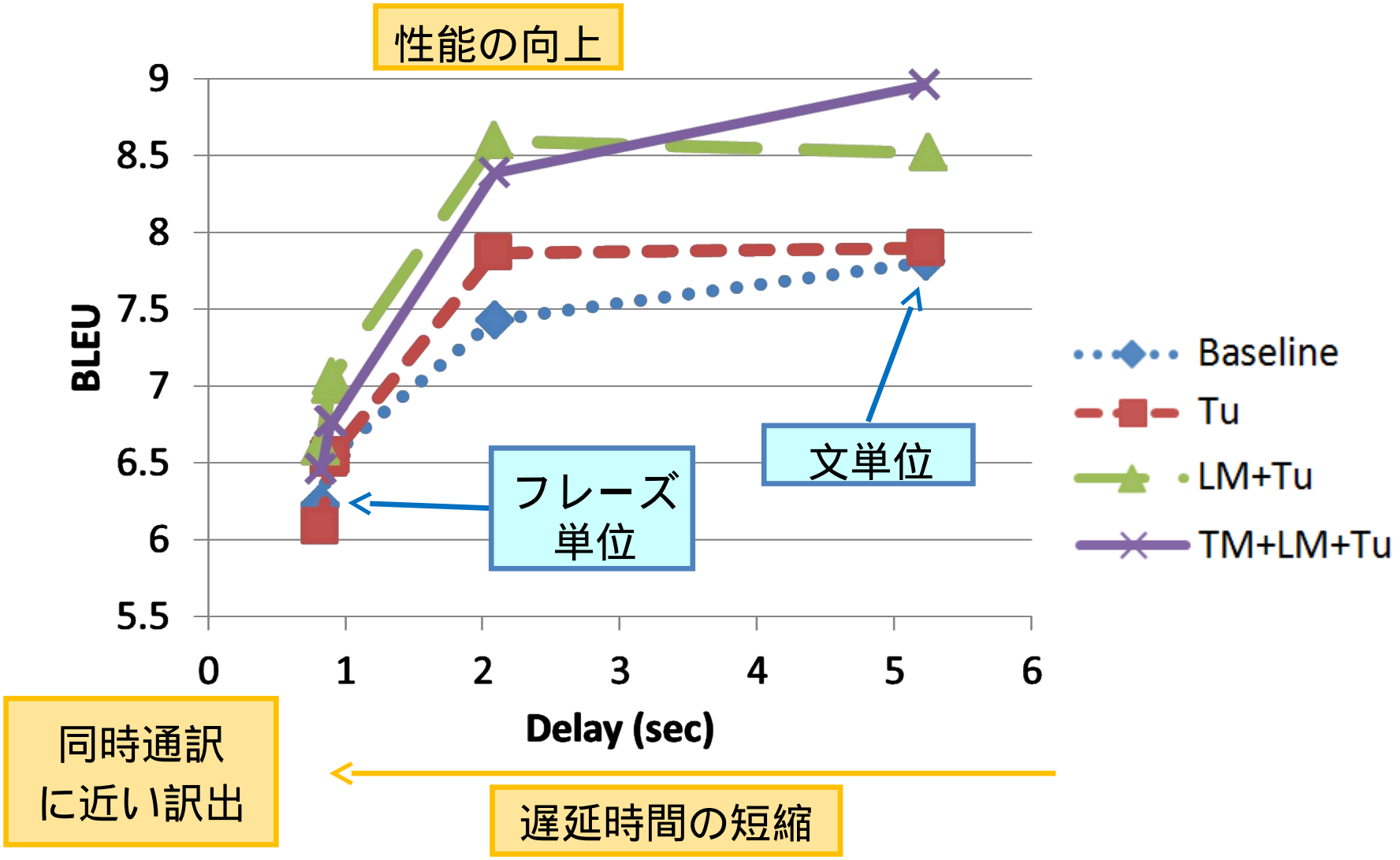


分野適応の技術を用いた通訳者の再現

- チューニング（Tu）
  - 同時通訳者の訳出結果に近づくようにパラメータが調節されることを期待
- 言語モデル（LM）：線形補間
  - 同時通訳者のような語順や語彙選択を期待
- 翻訳モデル（TM）：fill-up 法 [Bisazza+ 11]
  - LM と同様，同時通訳者のような語彙選択を期待

機械翻訳システムの学習における 3 つの過程に
同時通訳データを利用

通訳データを用いた評価実験



訳出の例

	例文
入力	If you look at in the context of the history you can see what this is doing
正解	過去から流れを見てみますと災害はこのように増えています
従来	見てみると歴史の中で見ることができますこれがやっていること
提案	では歴史の中で見ることができますこれがやっていること

- 単語数の減少
  - チューニングによって短いフレーズを好む パラメータに調整
- 翻訳結果の 25% の文が「で」から開始
  - 同時通訳者が次の文を待つまでの沈黙を回避

訳出タイミングの最適化
[小田 + 14]

訳出タイミング決定法の問題

すべてヒューリスティクスに基づく手法
音韻的情報、言語的情報　…

- 分割位置が翻訳精度に与える影響を考慮せず
- 翻訳器に対して分割位置が最適化されていない

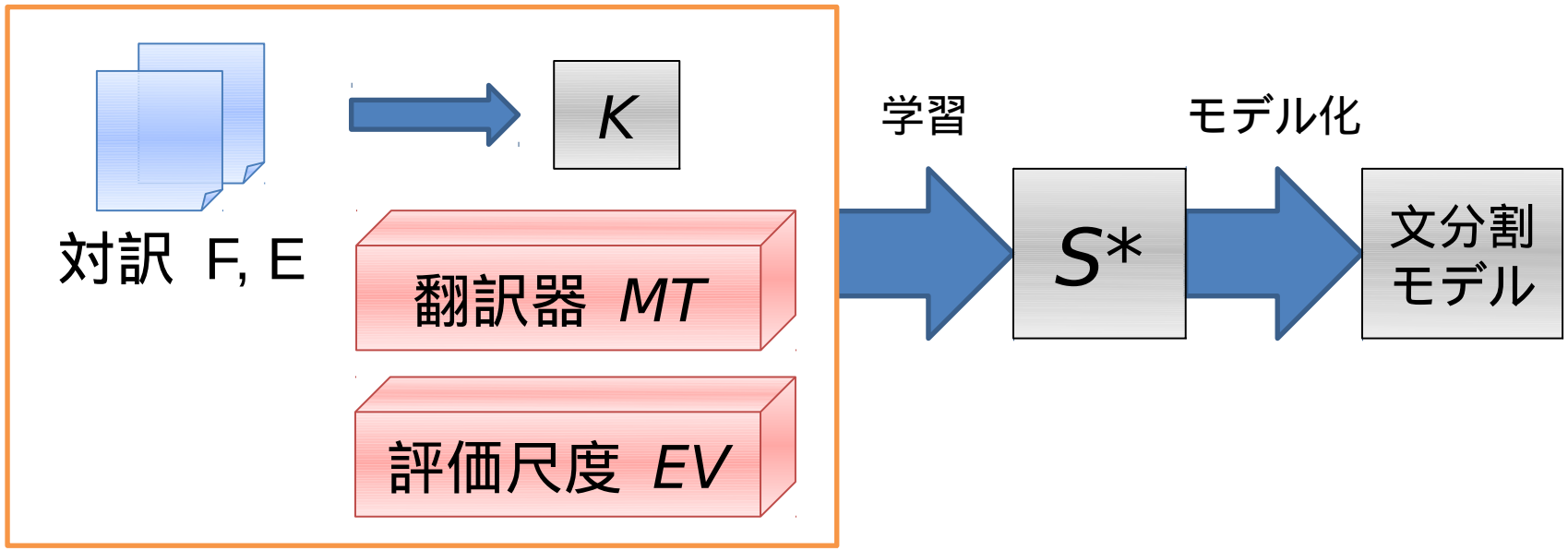
タイミングを最適化したい！

定式化

1. 学習データ（対訳コーパス）全体で分割する数 K を決定
   （パラメータ）
2. K 個の分割位置を学習データから選択

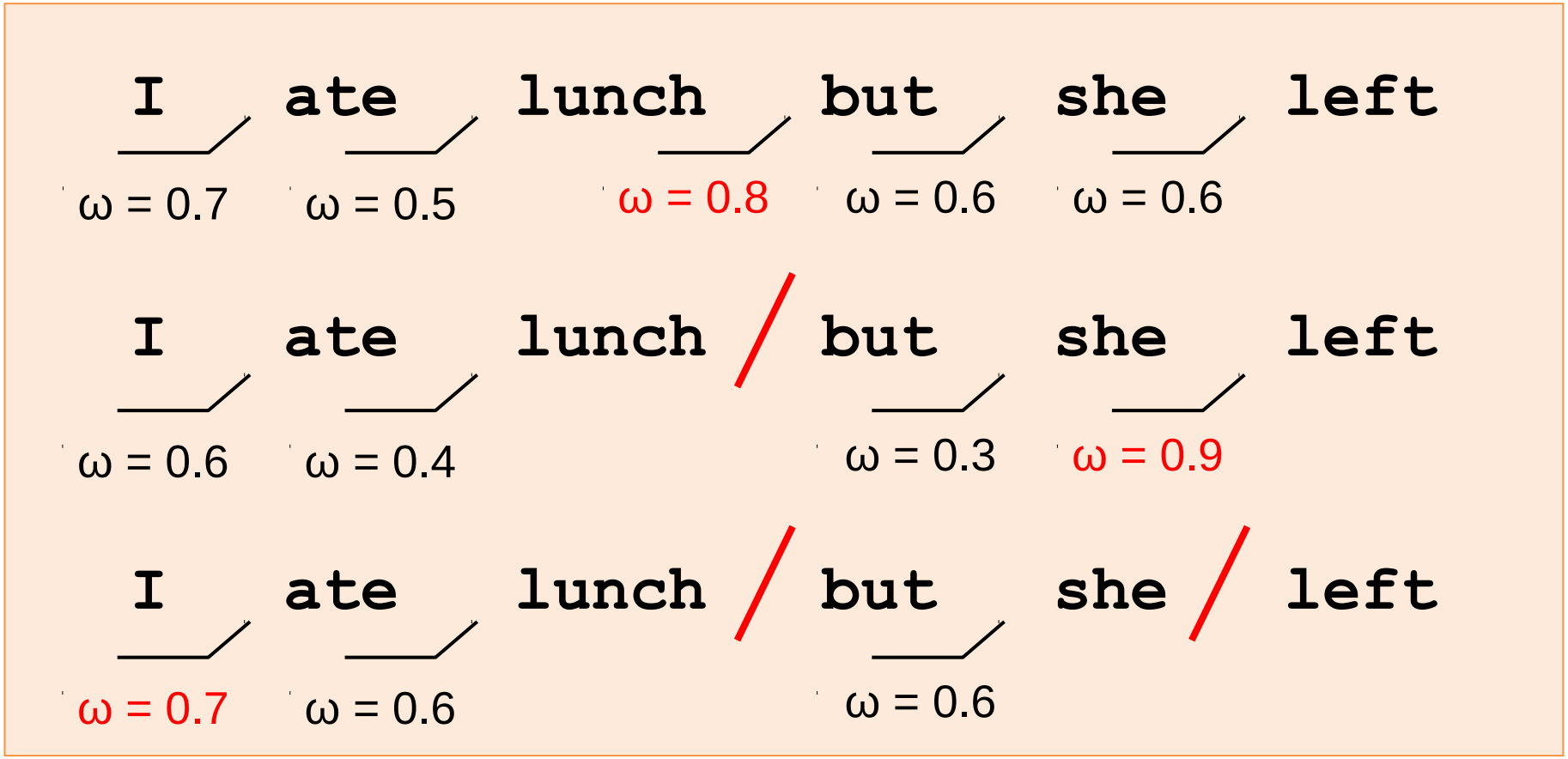
$$\mathcal{S}^* := \arg\max_{\mathcal{S}} \omega(\mathcal{S}|\mathcal{F}, \mathcal{E}, EV, MT)$$

3. 分割位置の素性をモデル化



手法 1: 貪欲法による分割

- 次の分割位置を決めるとき、今までに選んだ分割位置を保持
  (= 貪欲法 : greedy search)



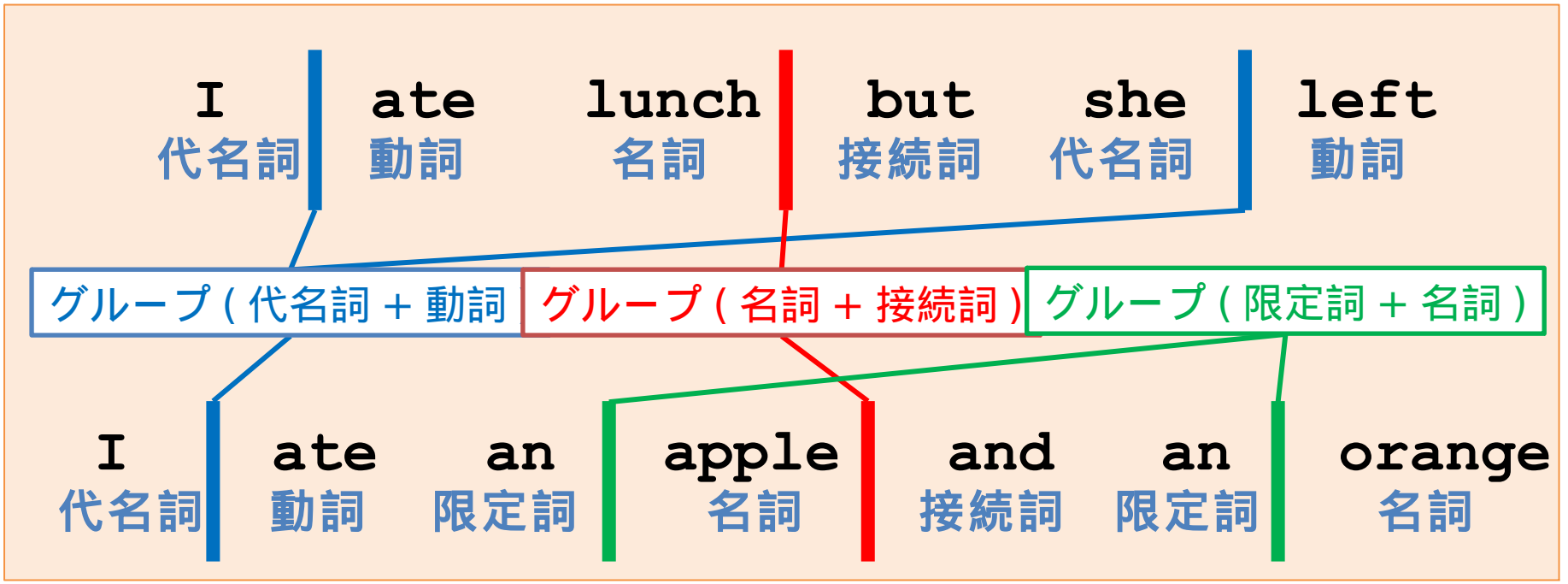
選ばれた分割位置の素性を SVM で学習

手法２：素性のグループ化

- ω は複雑な関数、ノイズが多い
  - 貪欲法では偶然 ω が良くなる分割位置で過学習

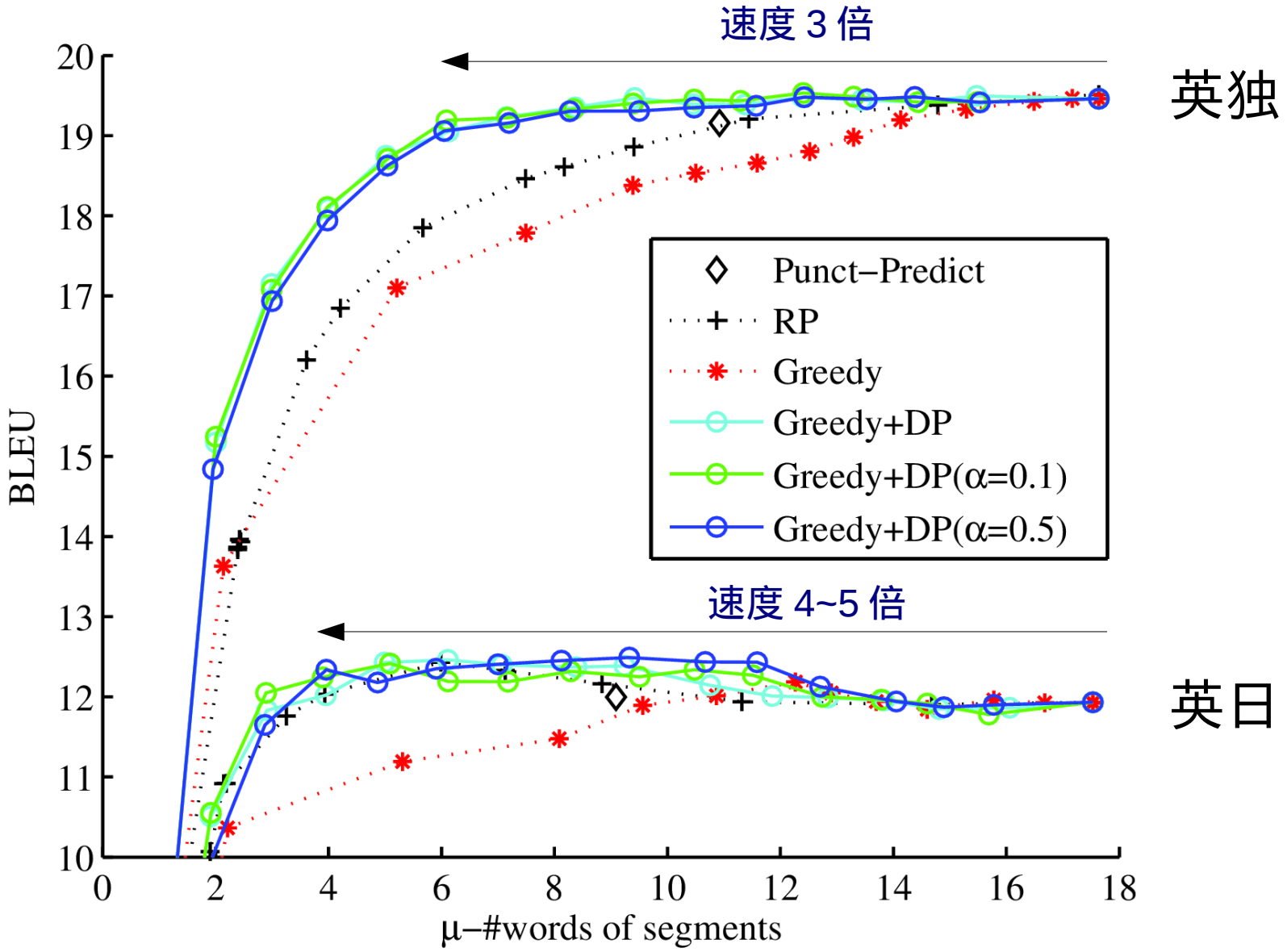
解決策　…　同じ素性を持つ分割位置をグループ化、同時に分割

例：前後の品詞

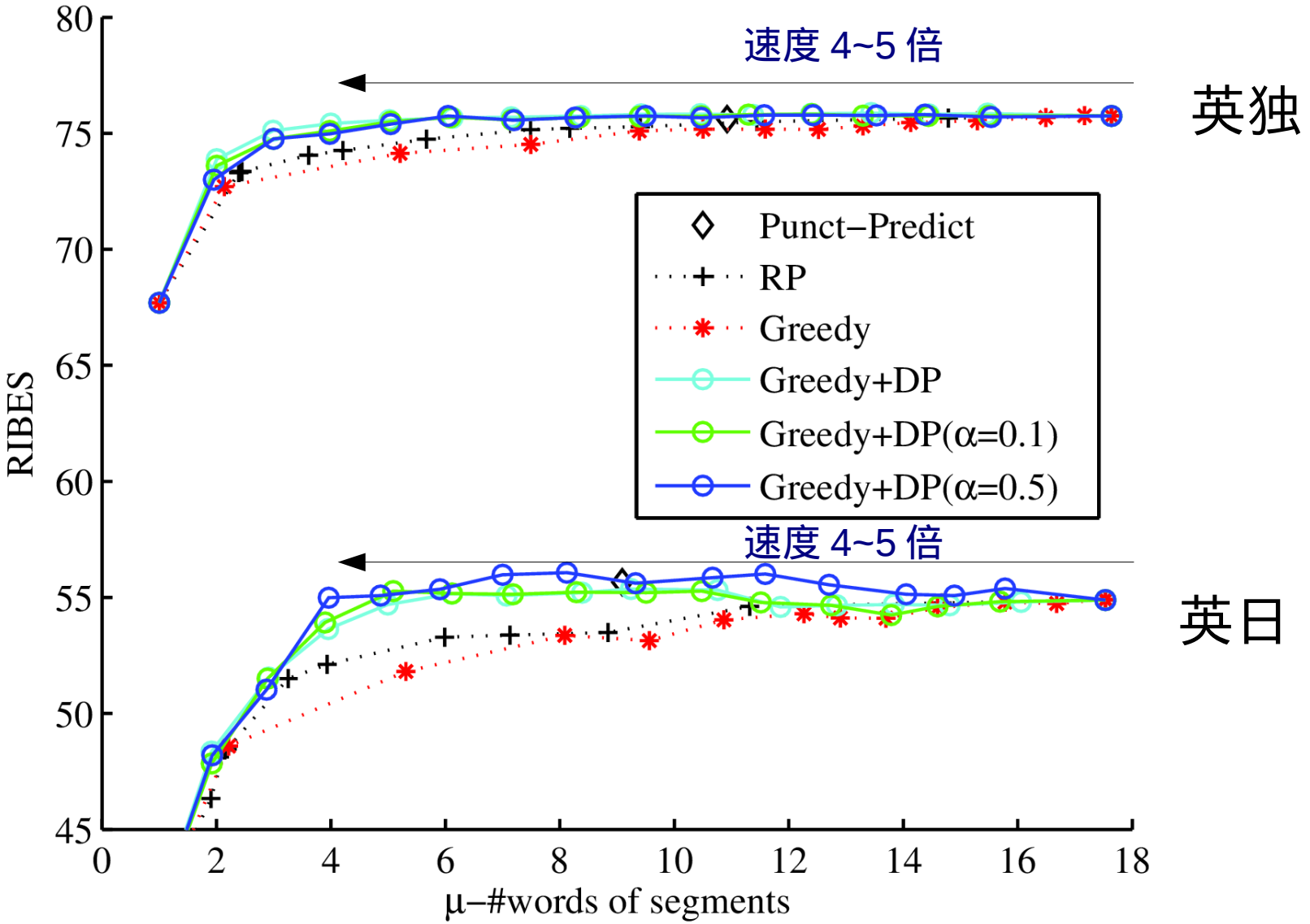


動的計画法 (DP) で探索、
探索で素性が得られるので　モデル化は不要
正則化の導入も可能

実験結果（BLEU）


速度 3 倍
速度 4~5 倍
英独
英日
BLEU
μ–#words of segments
Punct–Predict
RP
Greedy
Greedy+DP
Greedy+DP(α=0.1)
Greedy+DP(α=0.5)
20
19
18
17
16
15
14
13
12
11
10
0
2
4
6
8
10
12
14
16
18

実験結果 (RIBES)


速度 4~5 倍
英独
速度 4~5 倍
英日
RIBES
μ–#words of segments
Punct–Predict
RP
Greedy
Greedy+DP
Greedy+DP(α=0.1)
Greedy+DP(α=0.5)
80
75
70
65
60
55
50
45
0
2
4
6
8
10
12
14
16
18

声の特徴を翻訳する音声翻訳
[Kano+ 12, Kano+ 13]

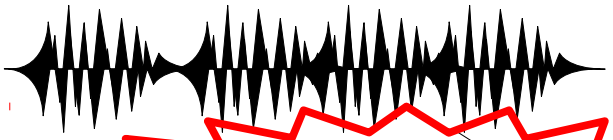
声の特徴は多く語る

映画の吹き替え

音声翻訳にかけると…

問題！

- 音声認識の時点で声の特徴が失われる…



音声認識

こんにちは、駅はどこですか？

機械翻訳

Hello, where is the station?

音声合成

解決策

- 声の特徴も翻訳する！

音声分析

日本語の声の特徴

音響翻訳

英語の特徴

音声認識

こんにちは、駅はどこですか？

機械翻訳

Hello, where is the station?

音声合成

実験題材：数字の強調

Hello, I'm Mike.
My membership number is 581.

No !,
my ID is
5**8**1.

Five **Eight** One
(強調)

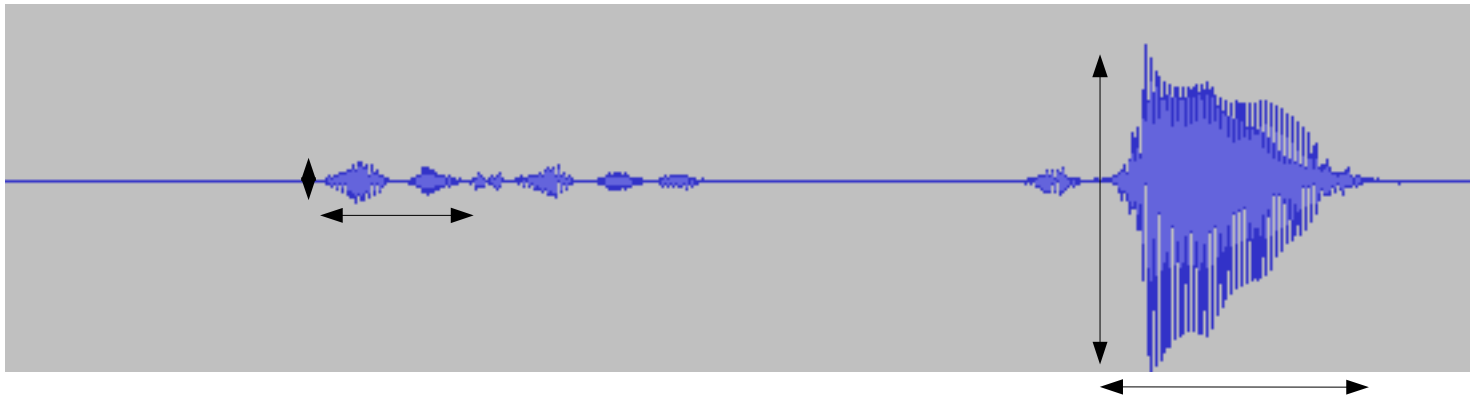
会員番号511 の
マイク様です
ね？

失礼しました .
*会員番号5**8**1 の*
マイク様ですね？

ご **はち** ご
(強調)

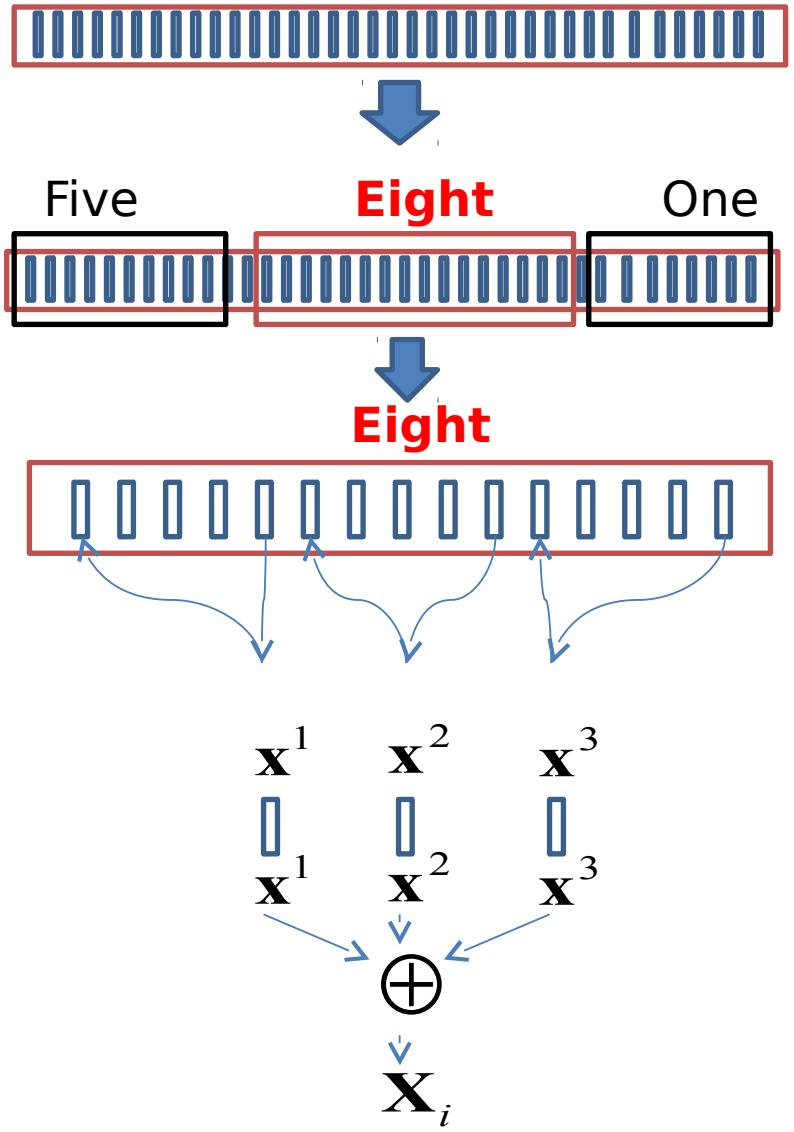
声の強調と特徴量

- 強調の時に、数字化できる声の特徴が変わる
- 特に、「継続長」と「パワー」が重要である



- つまり、これらの特徴量を翻訳すれば原言語へ反映できる！

声の特徴の認識



単語と、単語の始まり、真ん中、終わりの区間を認識

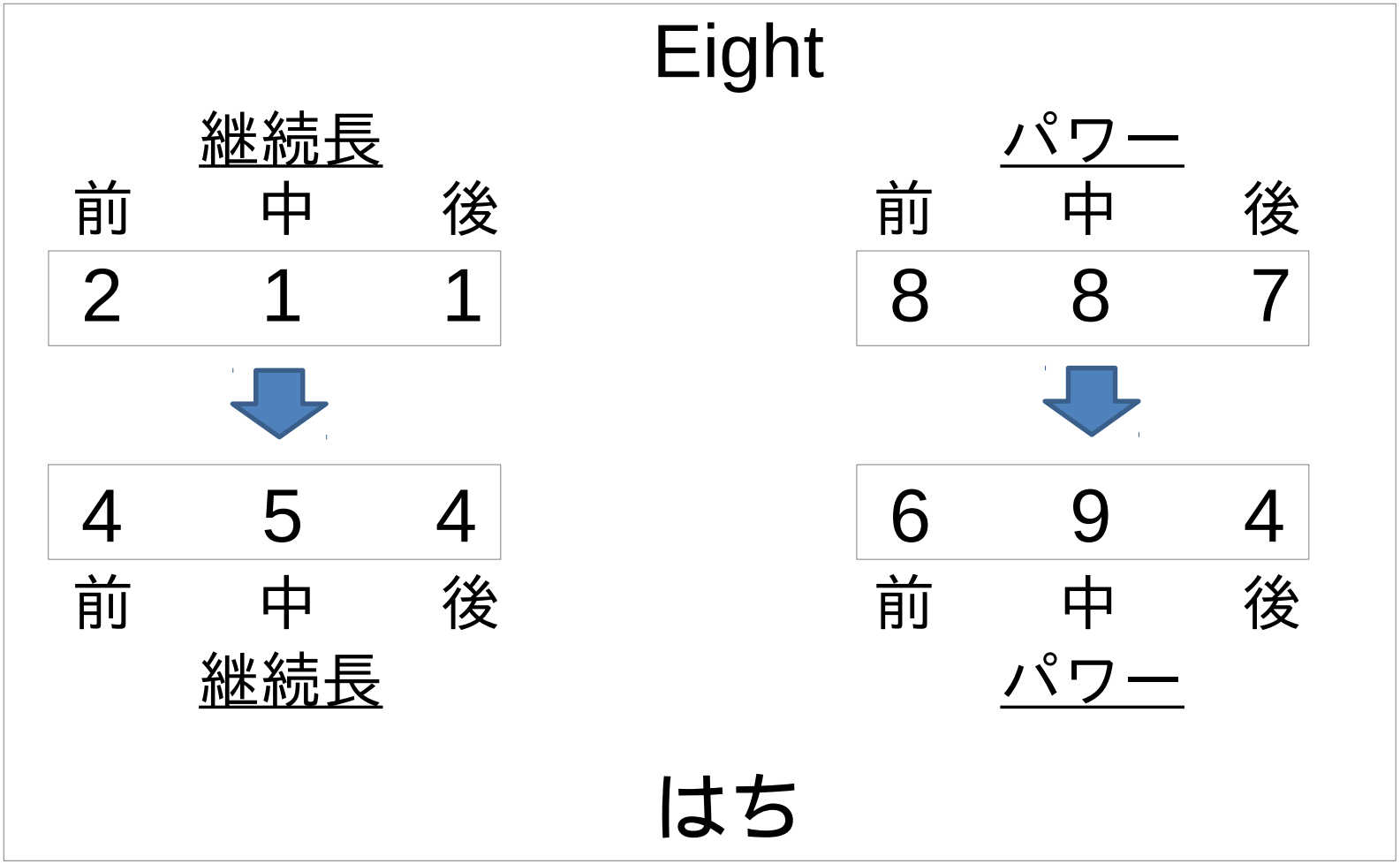
区間の長さ
→ 音声の継続長

区間の平均パワー
→ 音声のパワー

音声の特徴として扱う

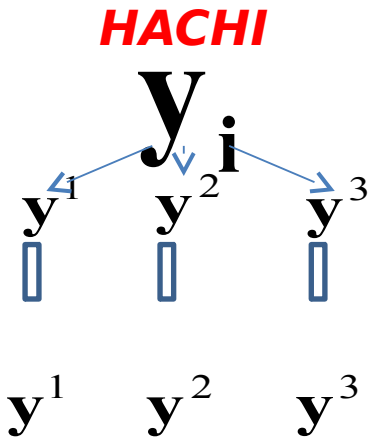
音声特徴量の変換

- これらの特徴量を原言語から目的言語へ変換



- 変換の関数をデータから学習

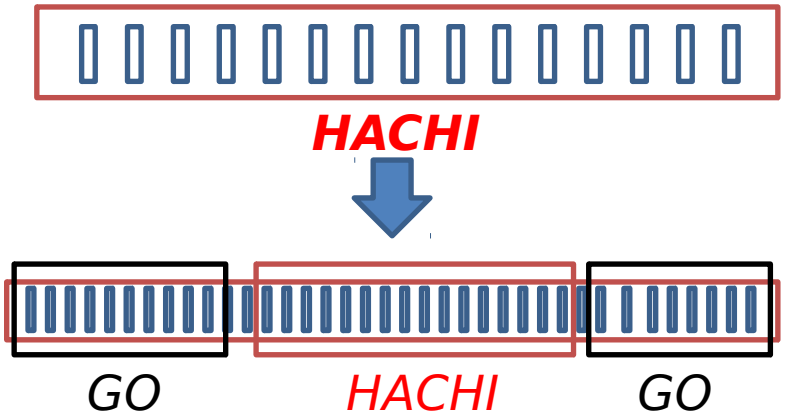
音声合成



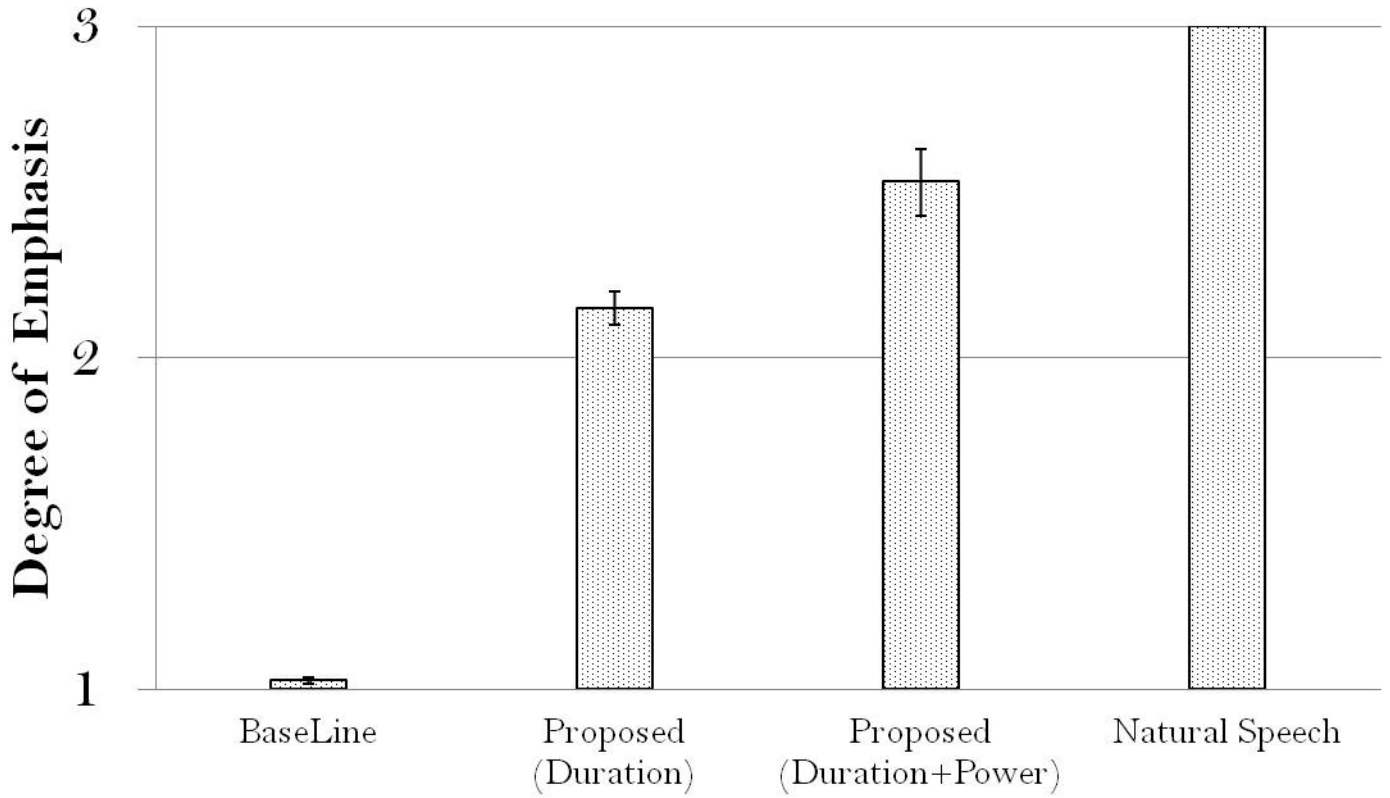
$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} P\left(\mathbf{O} \middle| \hat{\mathbf{J}}, \hat{\mathbf{Y}}\right)$$

$$subject \quad to \quad \mathbf{O} = \mathbf{MC}$$

声の特徴を合成時にも反映！



実験結果



- 継続長とパワーを反映することで強調を認識！

システムデモ

参考文献

参考文献

- [1] 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, 2012.
- [2] Arianna Bisazza, Nick Ruiz, and Marcello Federico. Fill-up versus interpolation methods for phrase-based smt adaptation. In Proc. IWSLT, pp. 136-143, 2011. 1
- [3] Christian Fugen, Alex Waibel, and Muntsin Kolss. Simultaneous translation of lectures and speeches. Machine Translation, Vol. 21, No. 4, pp. 209-252, 2007.
- [4] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Simple, lexicalized choice of translation timing for simultaneous speech translation. In Proc. 14th InterSpeech, 2013.
- [5] Roderick Jones. Conference interpreting explained, Vol. 6. St Jerome Pub, 2002.
- [6] Takatomo Kano, Sakriani Sakti, Shinnosuke Takamichi, Graham Neubig, Tomoki Toda, and Satoshi Nakamura. A method for translation of paralinguistic information. In Proc. IWSLT, Hong Kong, December 2012.
- [7] Takatomo Kano, Shinnosuke Takamichi, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura. Generalizing continuousspace translation of paralinguistic information. In Proc. 14th InterSpeech, 2013.
- [8] Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan. Segmentation strategies for streaming speech translation. In Proc. NAACL, pp. 230-238, 2013.
- [9] Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Constructing a speech translation system using simultaneous interpretation data. In Proc. IWSLT, 2013.
- [10] Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Collection of a simultaneous translation corpus for comparative analysis. In Proc. LREC, 2014.
- [11] 小田悠介 , Graham Neubig, 清水宏晃 , Sakriani Sakti, 戸田智基 , 中村哲 . 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 . 言語処理学会 第 20 回年次大会発表論文集 , 2014.