

対話システム勉強会（第1回）

OpenNMT による 雑談対話機能の実装

大阪大学大学院情報科学研究科 博士後期課程2年
高山 隼矢

本講義の概要

目標：翻訳器構築ツール OpenNMT-py を用いた
Seq2Seq 型雑談対話システムの実装方法を理解する

➤Seq2Seq 型対話システム概説

- モデルアーキテクチャ
- 応用分野の紹介
- 対話システム特有の課題・簡単な解決策

➤OpenNMT-py を用いた雑談対話機能の実装

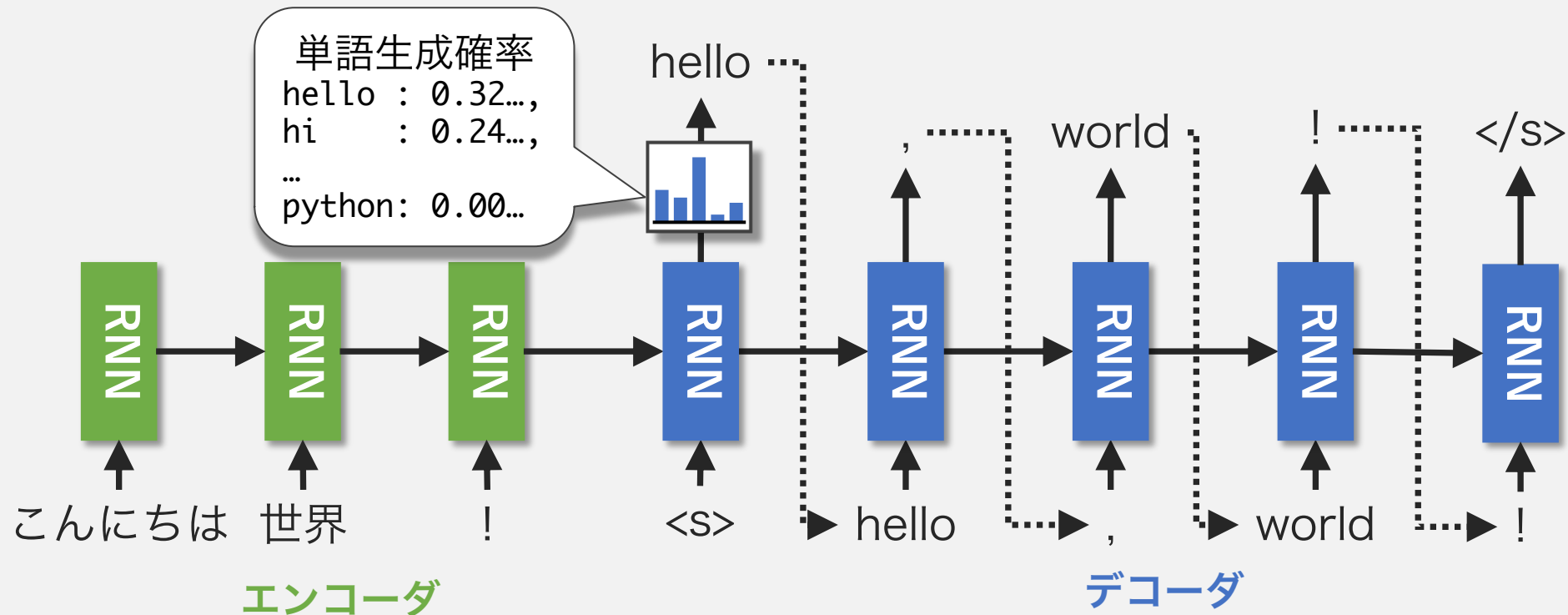
- おーぷん2ちゃんねるコーパスの前処理
- 対話システムの構築
- Rest API Server の構築

※ 「Python は多少わかるが自然言語処理は未経験」くらいの方々を対象とした講義です

Seq2Seq 型対話システム概説

Sequence-to-Sequence (Seq2Seq)

- テキスト X (例: 日本語文) を別のテキスト Y (例: 英語文) へ変換する機構
- X が与えられたとき Y が生成される確率を最大化するように学習
- 学習には大量の平行コーパス ($X \rightarrow Y$ の実例を収録したデータ) が必要



Seq2Seq の応用例

➤機械翻訳：原言語文 -> 目標言語文



➤自動要約：オリジナル文 -> 要約文（見出し文）

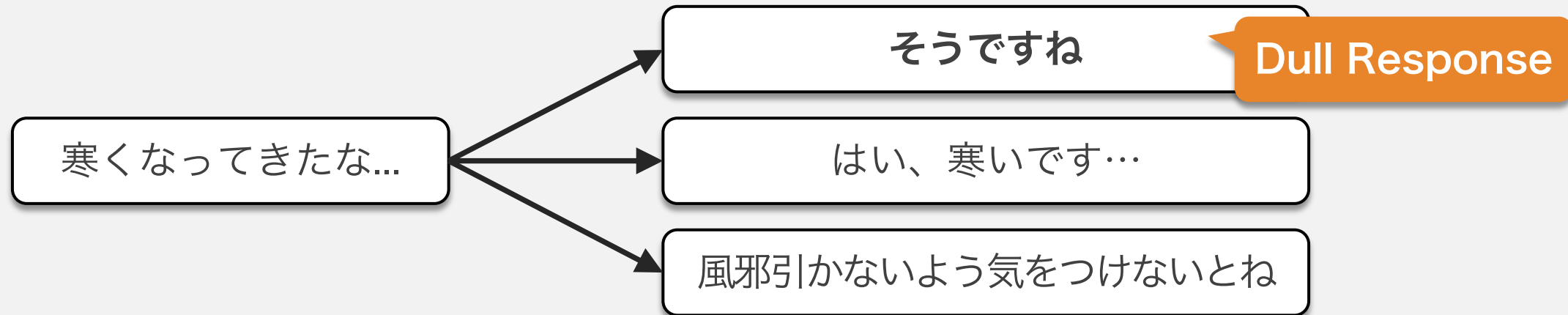


➤対話システム：発話文 -> 応答文



対話システム特有の課題: Dull Response

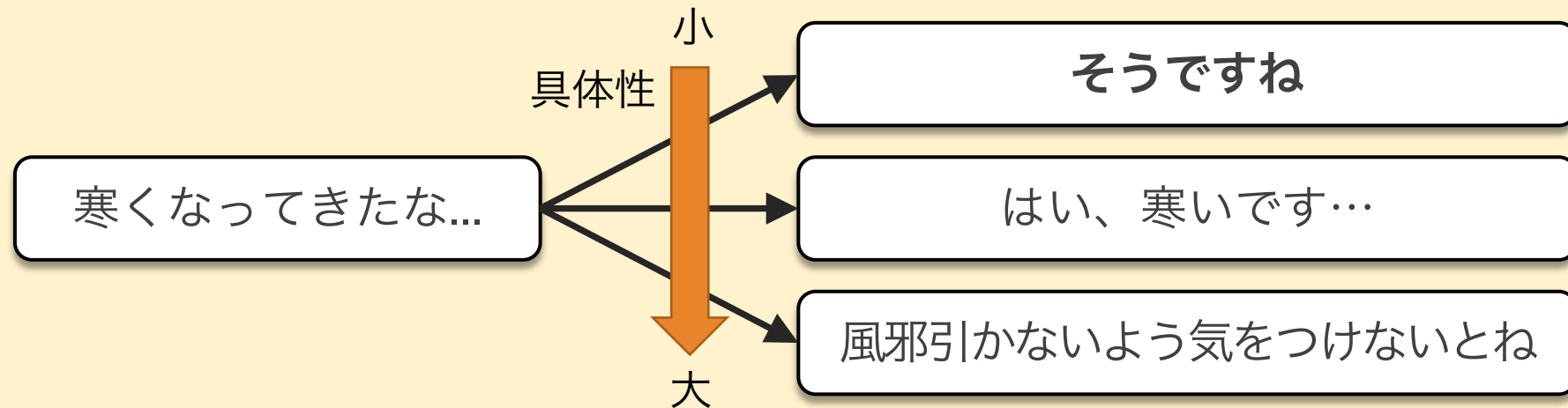
- 実際の対話では発話 X に対して可能な応答が Y 以外に複数存在
→ 人間は状況に応じて応答を使い分ける
- Seq2Seq 型アーキテクチャでは Y が生成される確率を最大化
→ 応答の使い分けを学習できず
高頻度で無難な応答 (Dull Response) ばかりを生成してしまう [1]



[宣伝] 我々の取り組み

発話に対する応答の具体性を明示的に制御できる応答生成手法の構築

1. J. Takayama, Y. Arase. "**Consistent Response Generation with Controlled Specificity**". Findings of EMNLP2020 (to appear)
2. 高山, 荒瀬. "**Distant Supervision** を用いた情報量を制御可能な雑談応答生成". NLP2020



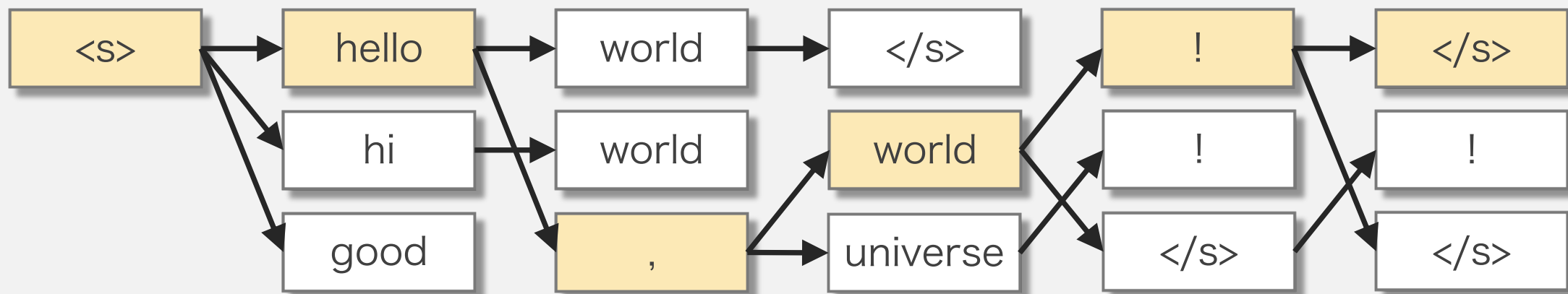
OpenNMT でもすぐに試せる Dull Response 問題対策： デコード方法を変えてみよう

➤デフォルト設定：Beam Search デコード

機械翻訳等，多くの文変換タスクにおける標準的なデコード法

高確率な単語 K 件を保持して幅優先探索し一番確率の高い文を採用

→生成確率が高い文を効率良く生成できる



翻訳の場合

生成確率の高い文 == (だいたい) 良い翻訳

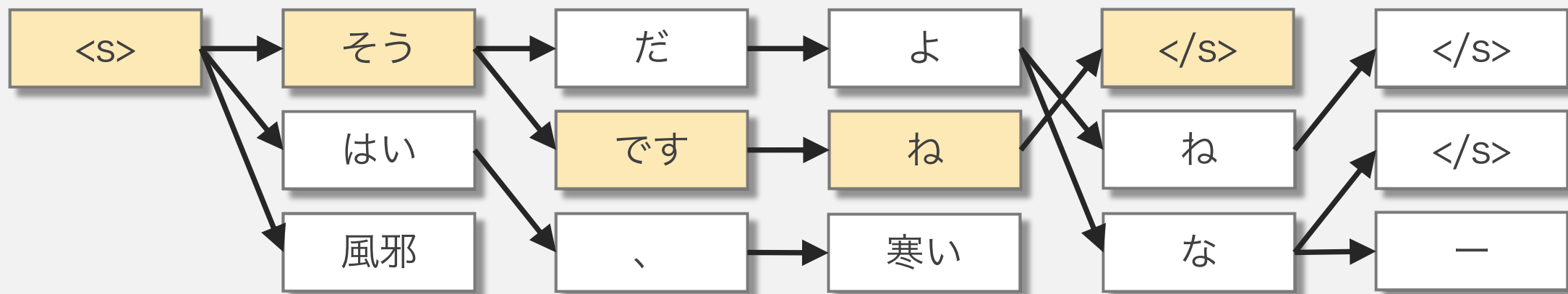
OpenNMT でもすぐに試せる Dull Response 問題対策： デコード方法を変えてみよう

➤デフォルト設定：Beam Search デコード

機械翻訳等，多くの文変換タスクにおける標準的なデコード法

高確率な単語 K 件を保持して幅優先探索し一番確率の高い文を採用

→生成確率が高い文を効率良く生成できる



対話の場合

生成確率の高い文 == (わりと) Dull Response

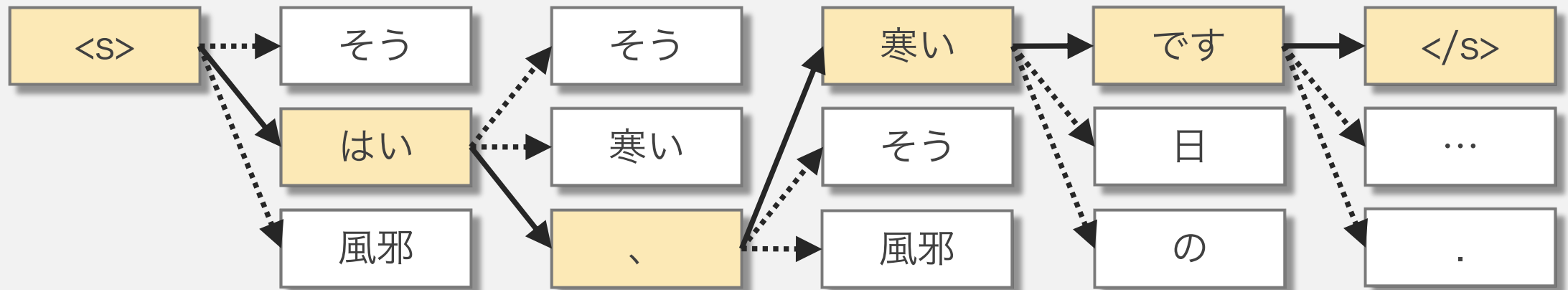
OpenNMT でもすぐに試せる Dull Response 問題対策： Top-K Sampling デコード

➤ オプション設定：Top-K Sampling デコード

単語選択にある程度の偶発性を持たせる手法

毎回確率が高い上位 K 件の単語を（確率に従って）ランダムに選択

→ 文の生成確率を"ほどほど"に考慮しつつ，無難な応答以外も出力できる



OpenNMT-py を用いた 雑談対話機能の実装

OpenNMT-py を用いた雑談対話機能の実装

➤OpenNMT-py とは

- Seq2Seq 型の機械翻訳器を構築するためのツール
- コマンドライン上で数回コマンドを入力するだけで翻訳器を構築可能
- 近年の機械翻訳の主流となっている Transformer モデルも実装可能

➤雑談対話機能の実装

- 以降の説明は基本的に Colab ベースで行います
- 自前環境で試したい方は Github リポジトリもご覧ください
https://github.com/junya-takayama/open2ch_chatbot_openNMT