教師なしニューラル単語分割を用いた分散表現獲得

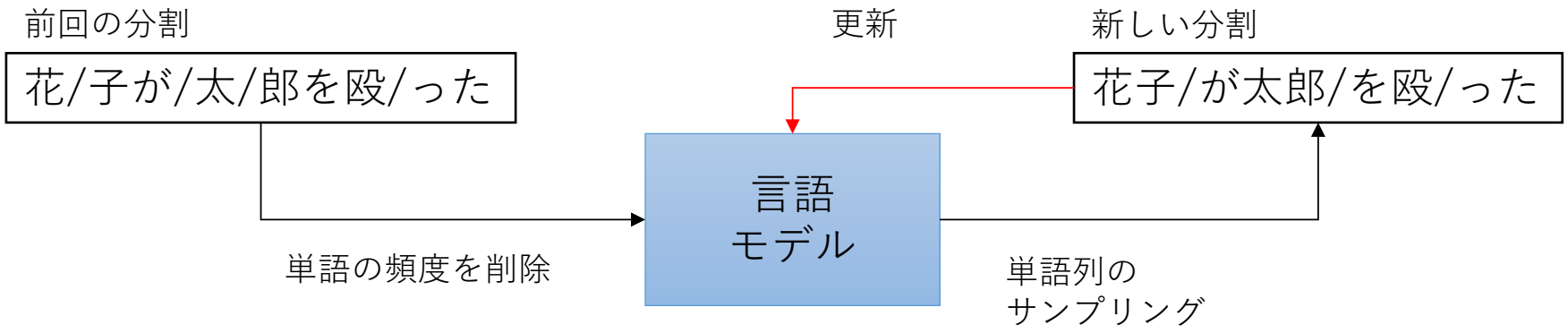
平岡達也(M2)，進藤裕之，松本裕治
奈良先端科学技術大学院大学（NAIST）

発表概要

- ニューラル言語モデルを用いた教師なし単語分割器の提案
  - 軽量なニューラル言語モデルを用いる
    - ニューラルN-gram言語モデル
  - 現実的な時間内で計算可能な設計
    - softmaxの近似
    - 動的計画法（Forward Filtering-Backward Sampling）によるサンプリング
- 分割の過程で得られる分散表現を用いた学習
  - 日本語文分類問題

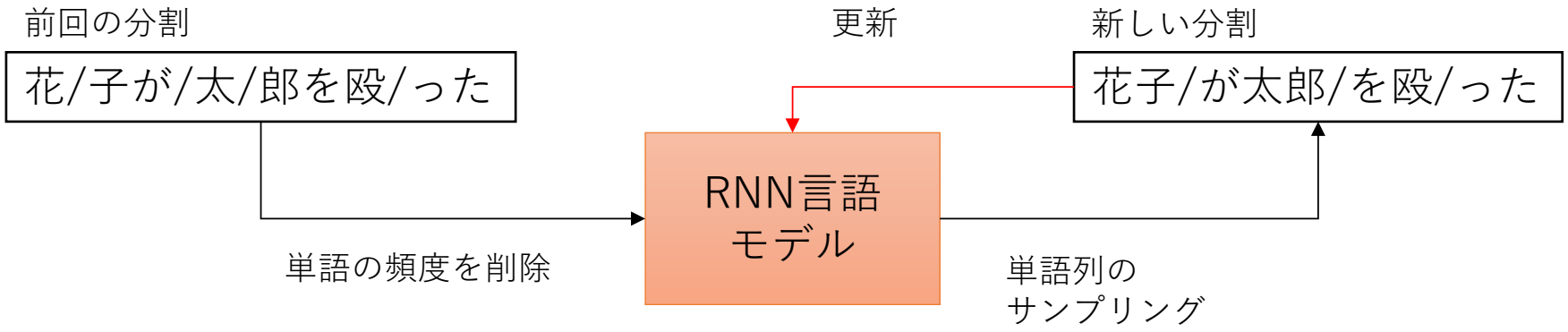
ギブスサンプリングによる教師なし単語分割

- 言語モデルの尤度が高くなるように分割状態を更新することで適切な単語分割が得られる.
  - NPYLM[持橋, 2009]



NLMを用いた教師なし単語分割

- [友利,2018]
  - 言語モデルとして文字レベル/単語レベルのRNN言語モデルを用いる
    - 単語頻度を管理し，頻度1以上の単語に分散表現を与えることで計算可能
    - 強化学習を用い，前から一単語ずつサンプリング



問題点

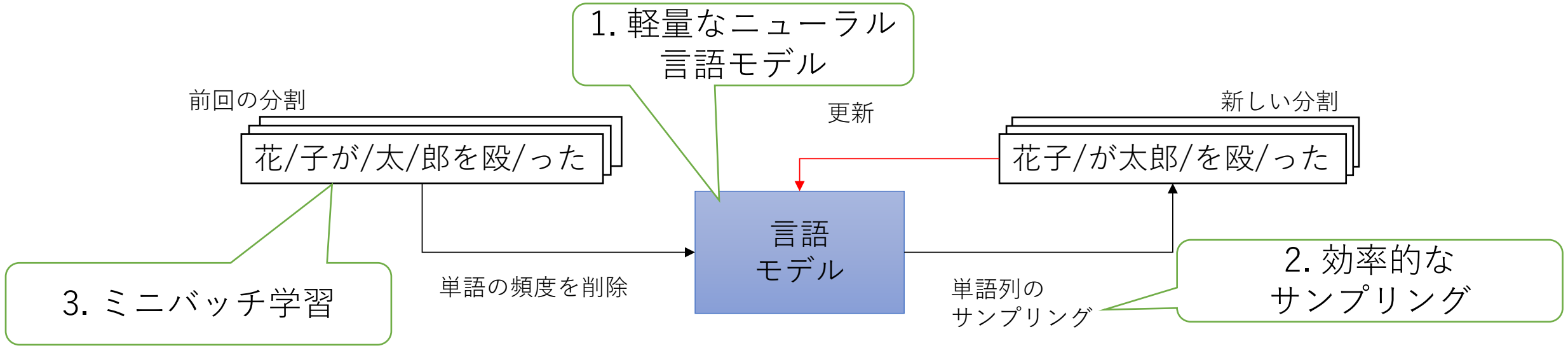
- 高い計算コスト
  - 単語レベル/文字レベルのRNN言語モデルを用いた計算
  - サンプリング時の報酬計算

- 局所解に陥る設計
  - モンテカルロ探索の回数を十分に大きく取る必要があるが,
    計算コストが高いため実現不可能

- **現実的な時間で計算可能なモデルを設計する必要がある**

提案手法の特徴

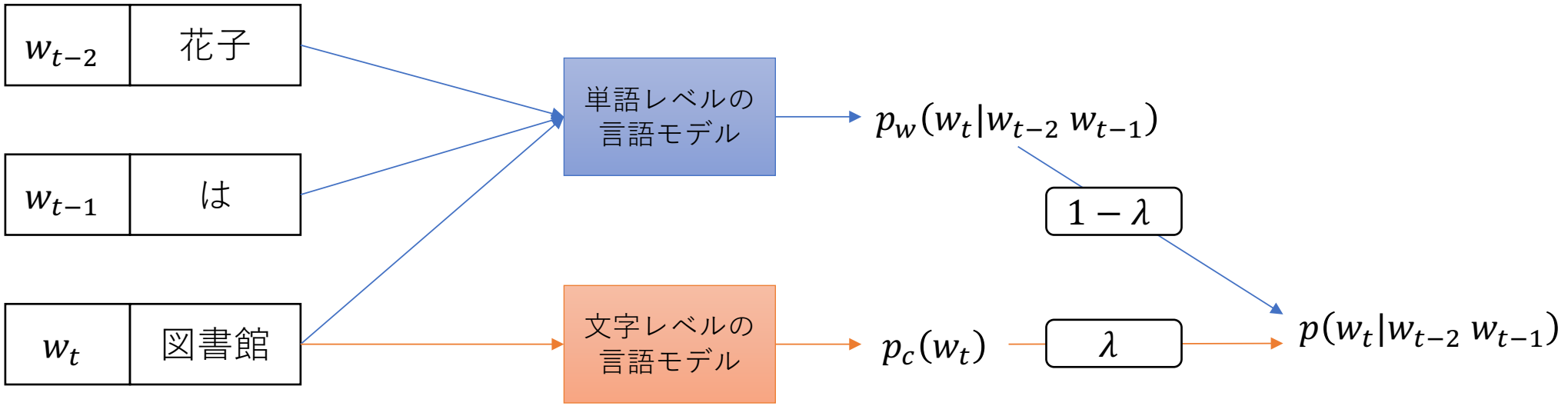
- 1. 計算コストの低い言語モデルを用いる
- 2. 動的計画法による効率的なサンプリングを行う
- 3. 複数文に対するミニバッチ学習を行う



言語モデル

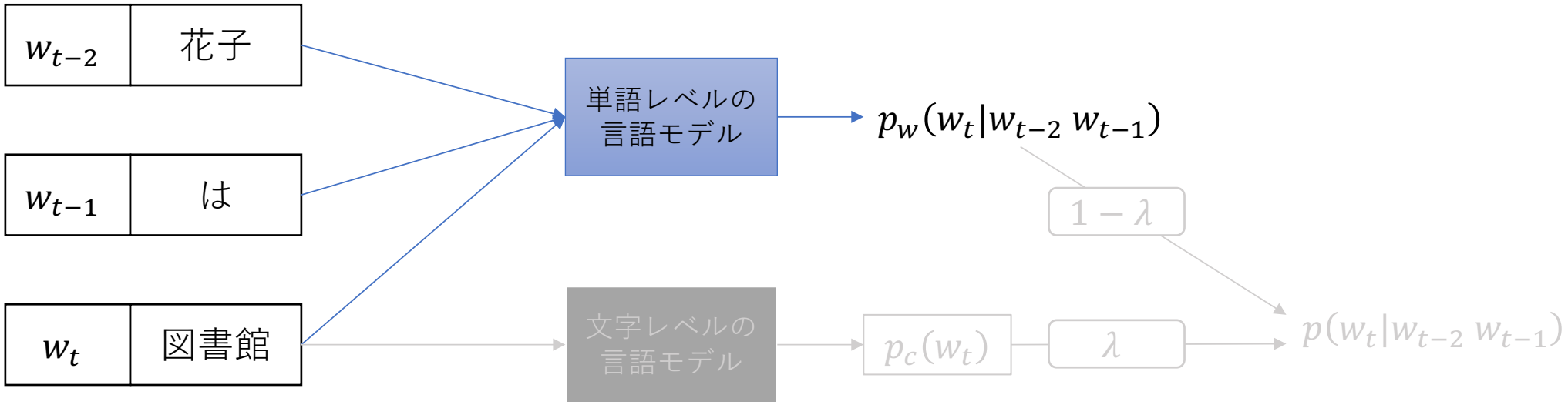
- 文字レベル/単語レベルの言語モデルの重み付き平均

$$p(w_t|w_{t-2}\ w_{t-1}) = (1-\lambda)p_w(w_t|w_{t-2}\ w_{t-1}) + \lambda p_c(w_t)$$



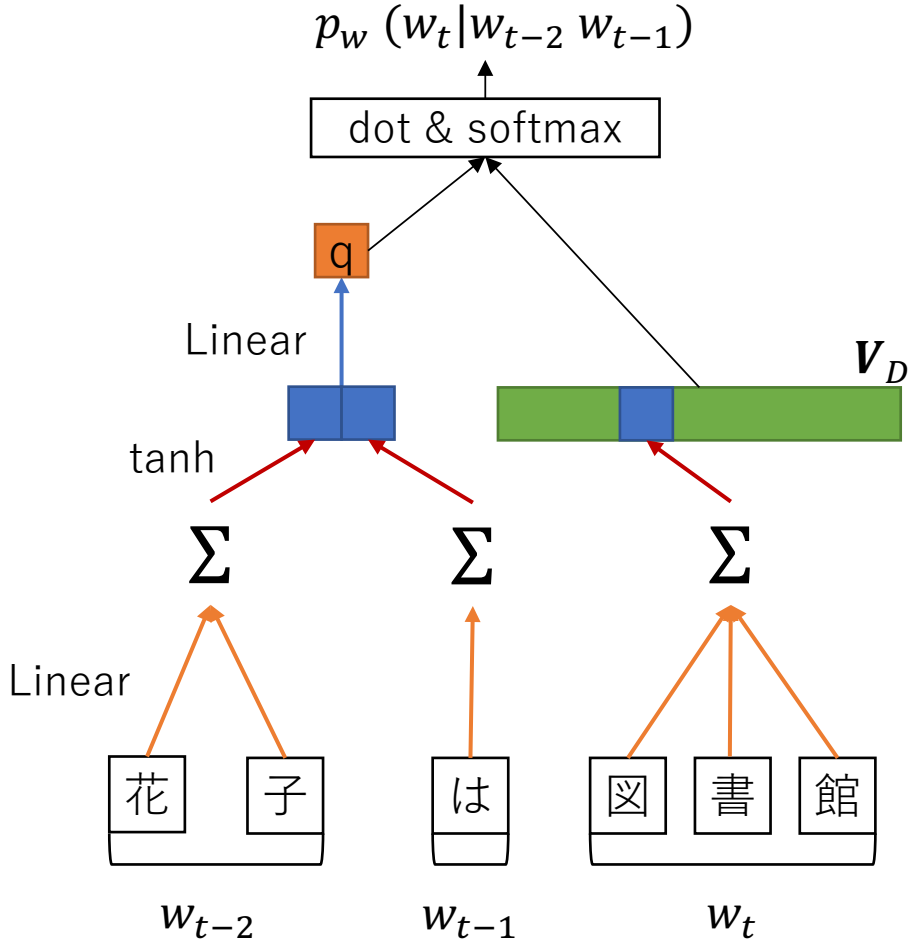
単語レベルの言語モデル

- ニューラル3-gram言語モデル
  - 直前2単語から文脈ベクトルを作り，次の単語の予測確率を計算



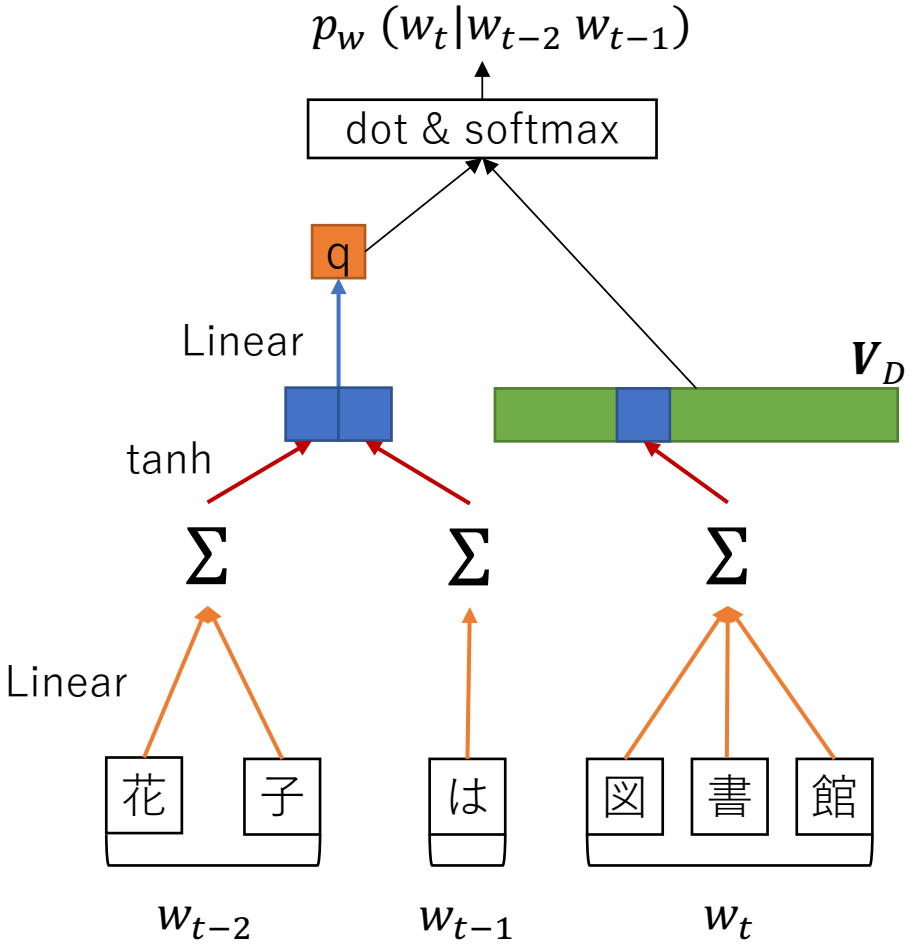
単語レベルの言語モデル

- 文脈ベクトル
  - $q = \tanh\left(\boldsymbol{W}\left(v_{w_{t-2}}; v_{w_{t-1}}\right) + b\right)$
- 単語確率の計算
  - $p(w_t|w_{t-2}w_{t-1}) = \mathrm{softmax}(q^T \boldsymbol{V}_D)_{v_{w_t}}$
  - 3-gramを構成する単語が現在の語彙に含まれない場合は確率を0にする



単語レベルの言語モデル

- 単語ベクトルの計算
  - [Cai, 2016]
    - 単語$w = c_1 \ \ldots c_i$
    - $v_w = \tanh(\sum_i \boldsymbol{W}_i v_{c_i} + b_i)$
    - 単語の最大長を決めることで,
      高速に文字ベクトルから単語ベクトルを
      計算可能



単語レベルの言語モデル

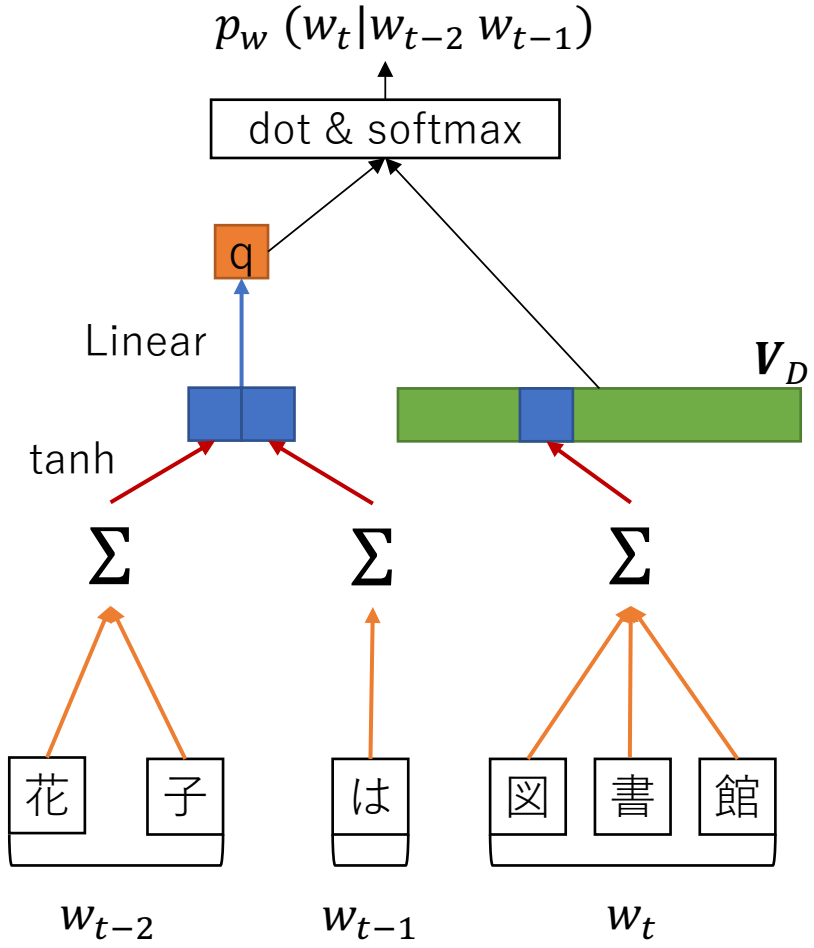
- ギブスサンプリングを行う上で,
  語彙サイズのsoftmaxがボトルネックになる

- $\mathrm{softmax}(q^T \boldsymbol{V}_D)$

$$= \frac{\exp(q^T v_w)}{\sum_{\hat{w}} \exp(q^T v_{\hat{w}})}$$

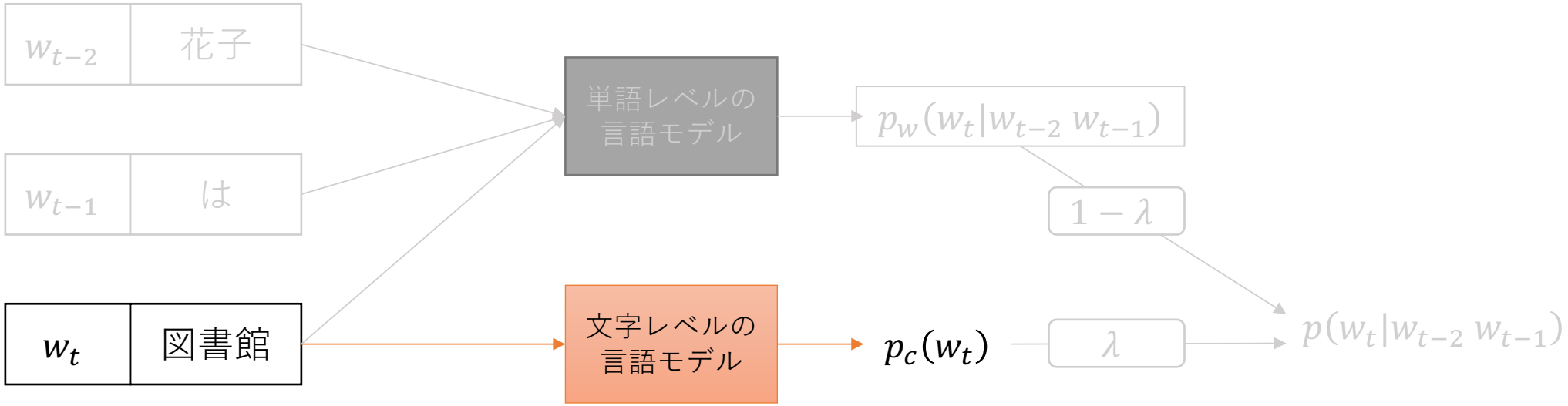
$$\simeq \frac{\exp(q^T v_w)}{\frac{|V_D|}{K} \sum_{w_1 \ldots w_K \sim uniform(V_D)} \exp\left(q^T v_{w_k}\right)}$$

- k個の単語によって分母を近似



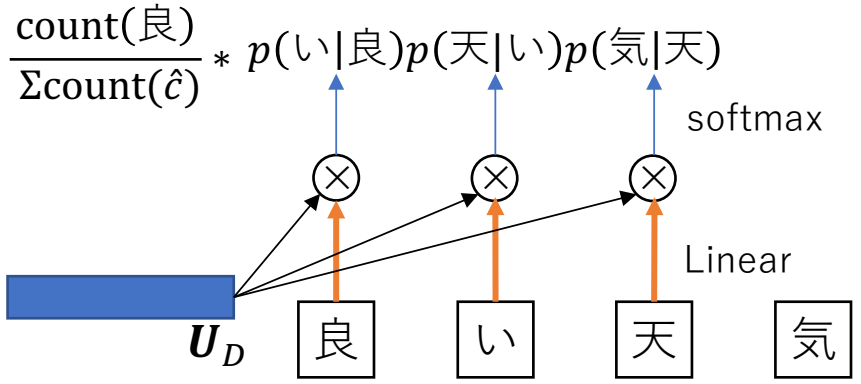
文字レベルの言語モデル

- 単語レベルの言語モデルp_wは0になりうる
  - 文字レベルの言語モデルによる平滑化を行う



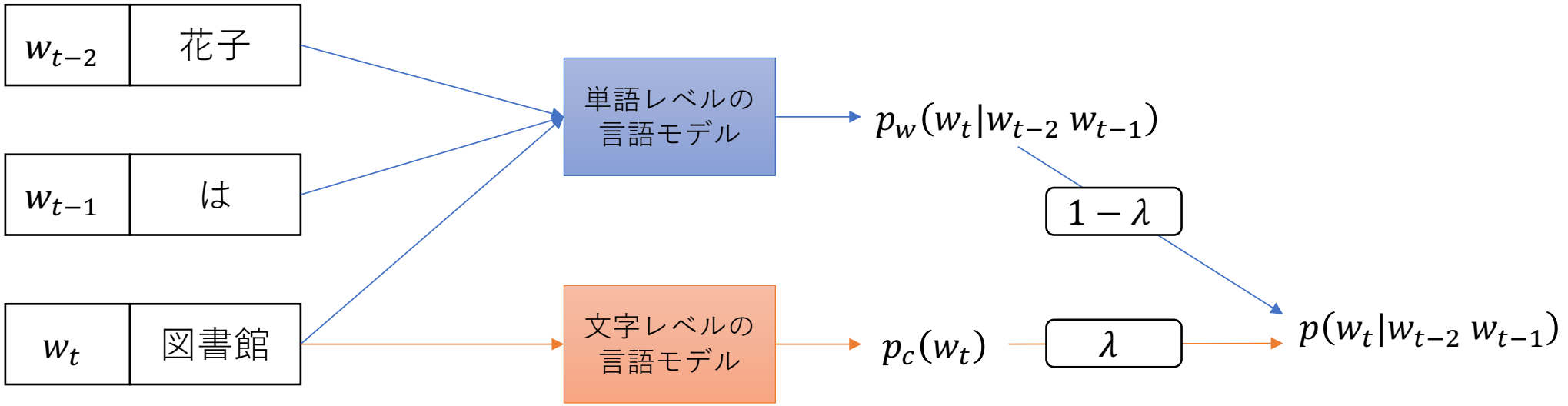
文字レベルの言語モデル

- 1文字目の離散文字ユニグラムと，2文字目以降のニューラル文字バイグラム確率による
  - $p(c_0 \dots c_M) = \frac{\mathrm{count}(c_0)}{\sum_{\hat{c}} \mathrm{count}(\hat{c})} \prod_{m=1}^{M} p_n(c_m|c_{m-1})$
    - BOWを用いると未知語の確率が非常に低くなる
  - $p(c_m|c_{m-1}) = \mathrm{softmax}(q_c \boldsymbol{U}_D)$
  - $q_C = \boldsymbol{W} v_{c_{m-1}} + b$



スムージング係数λ

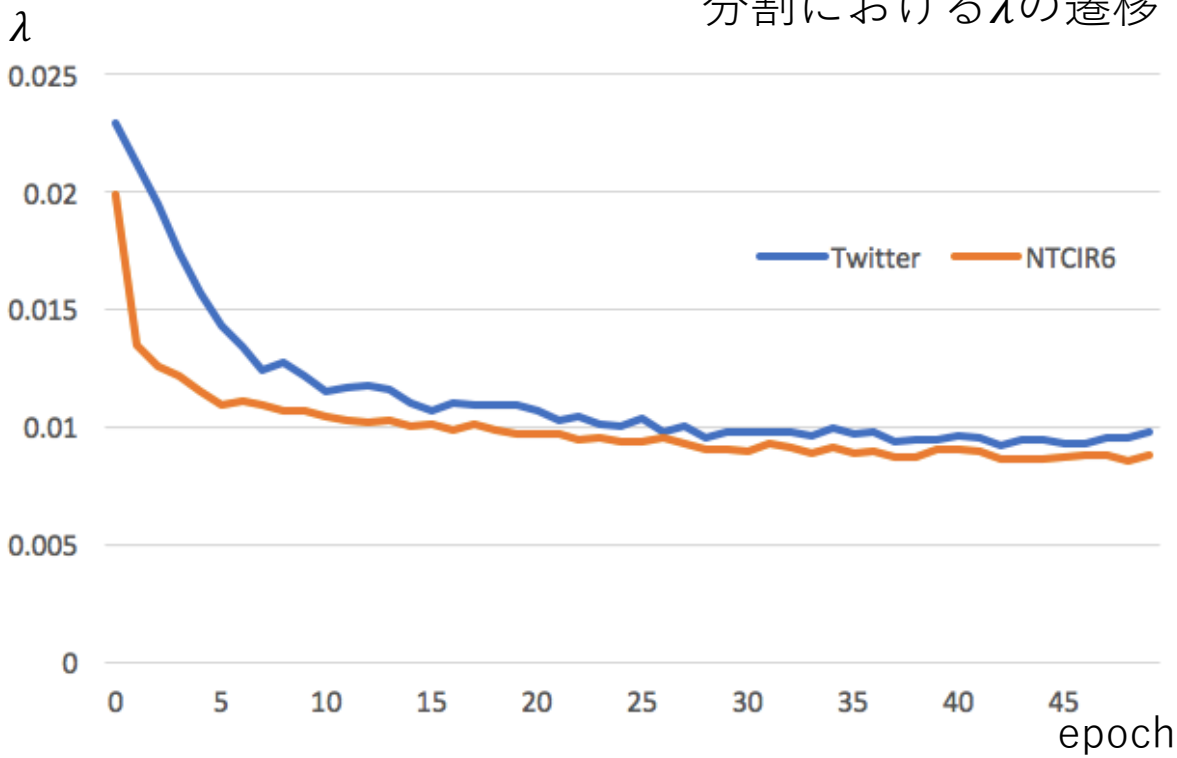
- λ が大きいほど，文字レベルの言語モデルに影響される
  - 既知な単語は単語レベルの言語モデルによる確率が高いため，
    λを大きくすることで未知語をサンプリングする確率を大きくできる．



スムージング係数λ

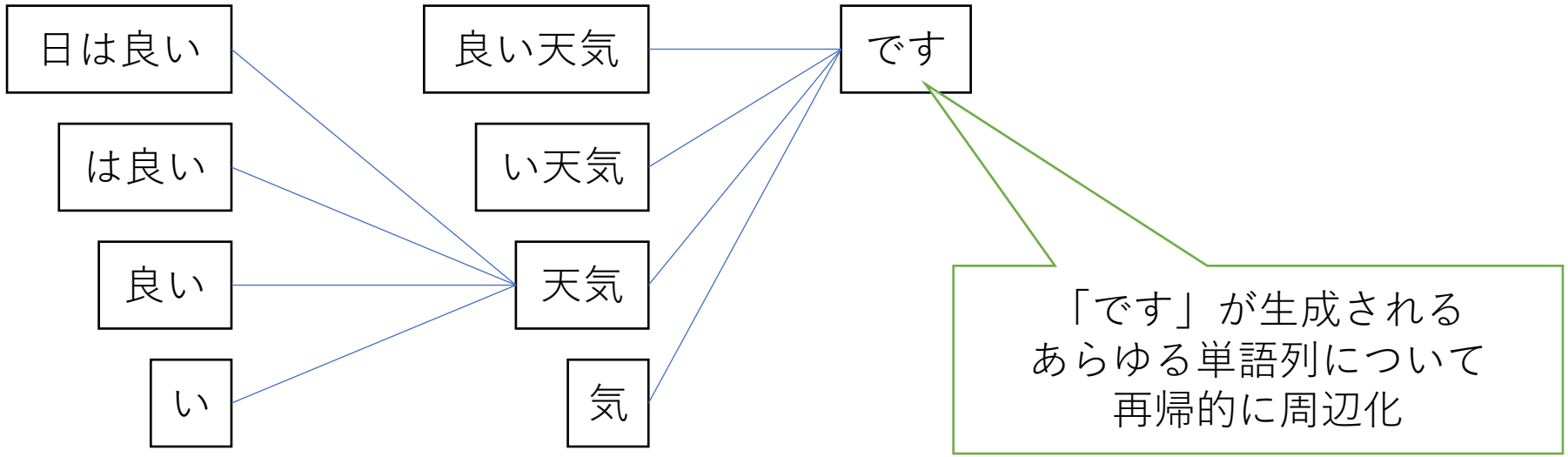
- λ は分割初期に大きめに取り，だんだん小さくする
- $\lambda = \frac{|\{w \in V_D \mid \mathrm{count}(w)=1\}|}{\sum_{\hat{w}} \mathrm{count}(\hat{w})}$
  - 現在の分割による語彙における頻度1の単語の割合によって設定

分割におけるλの遷移



サンプリング

- 動的計画法による効率的なサンプリング
  - Forward Filtering-Backward Sampling [Scott, 2002]
  - 前向き計算によって周辺化された単語確率を用いて
    後ろ向きにサンプリング.



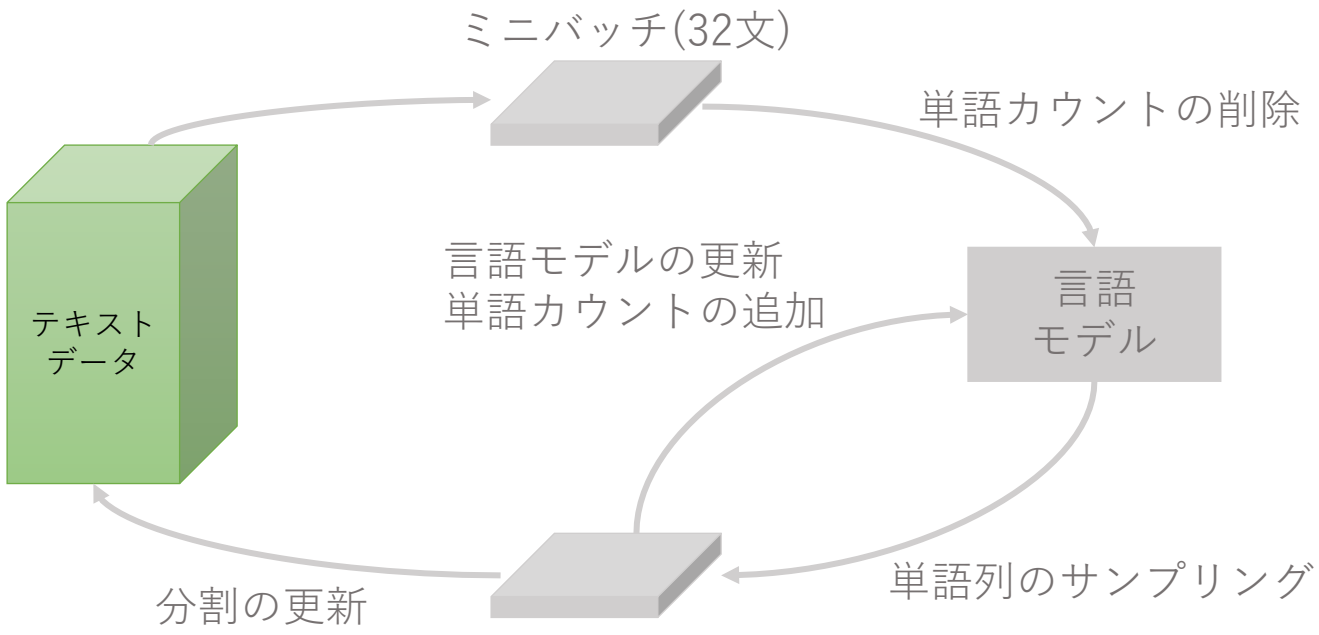
学習イテレーション

テキストをランダムに分割
```
for j <= maxEpoch:
  for each mini_batch:
    ミニバッチに含まれる単語のカウントを削除

    言語モデルから単語列をサンプリング

    新たな単語列で言語モデルを更新
    単語カウントの追加

    現在の単語分割を保存
```



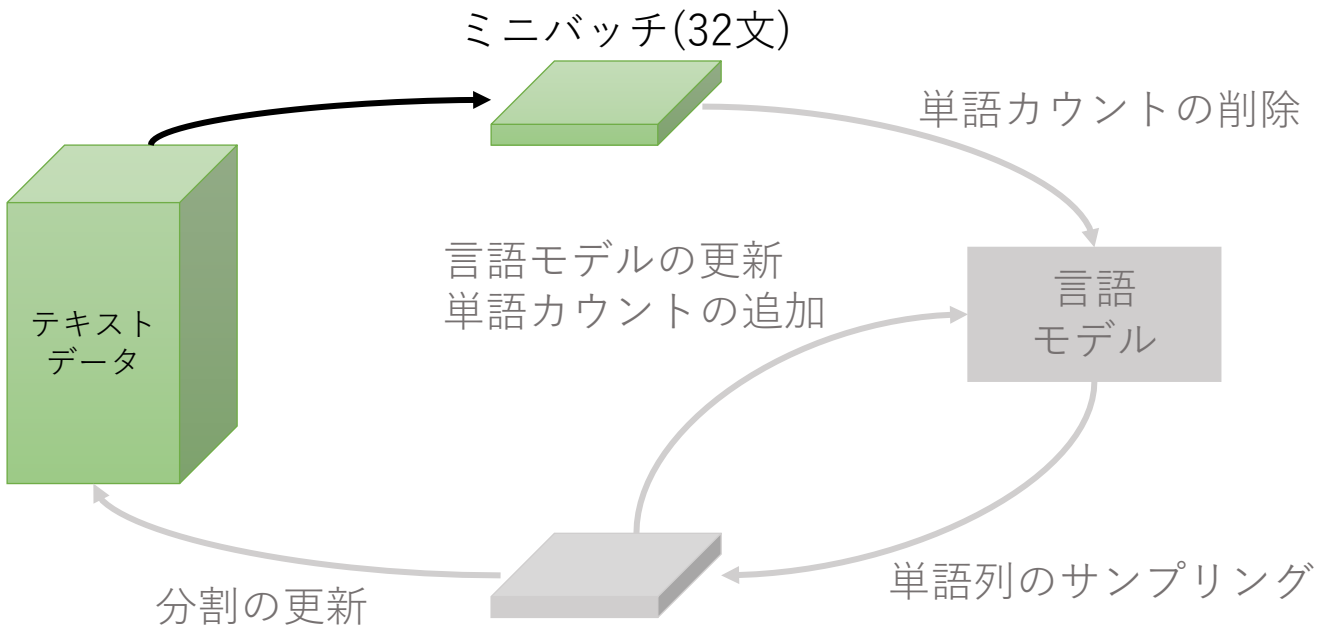
学習イテレーション

テキストをランダムに分割
```
for j <= maxEpoch:
  for each mini_batch:
```
ミニバッチに含まれる単語のカウントを削除

言語モデルから単語列をサンプリング

新たな単語列で言語モデルを更新
単語カウントの追加

現在の単語分割を保存



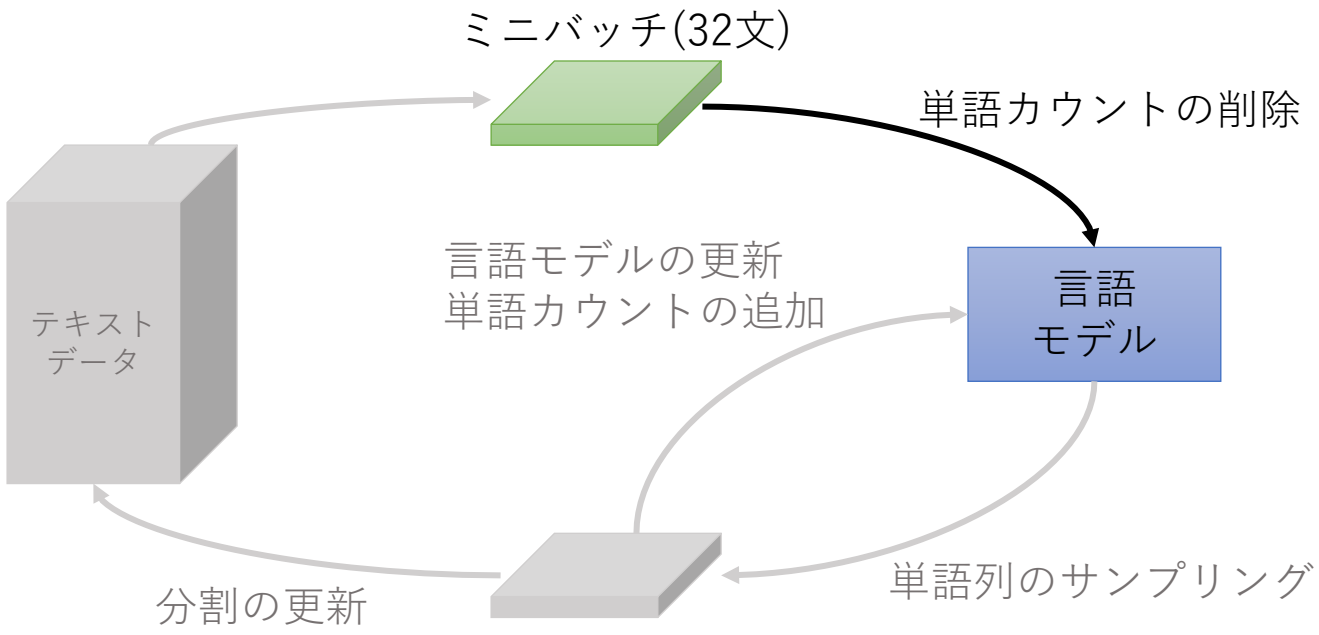
学習イテレーション

テキストをランダムに分割
```
for j <= maxEpoch:
  for each mini_batch:
```
ミニバッチに含まれる単語のカウントを削除

言語モデルから単語列をサンプリング

新たな単語列で言語モデルを更新
単語カウントの追加

現在の単語分割を保存



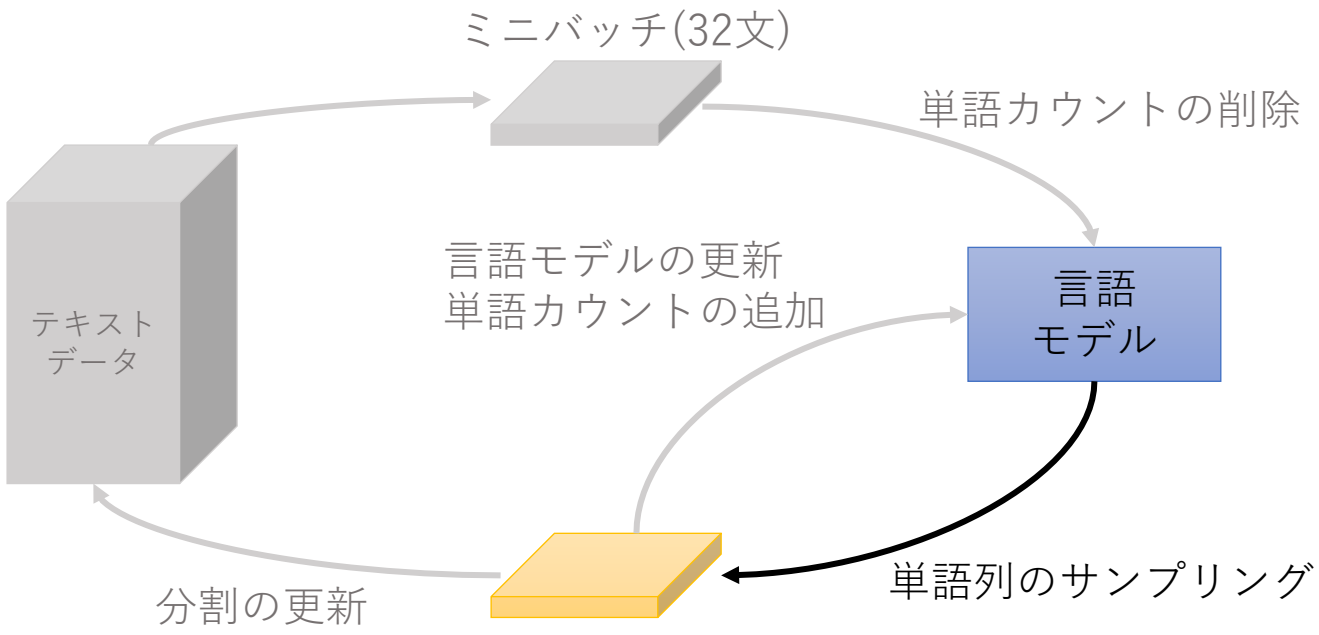
学習イテレーション

テキストをランダムに分割
```
for j <= maxEpoch:
  for each mini_batch:
```
ミニバッチに含まれる単語のカウントを削除

言語モデルから単語列をサンプリング

新たな単語列で言語モデルを更新
単語カウントの追加

現在の単語分割を保存



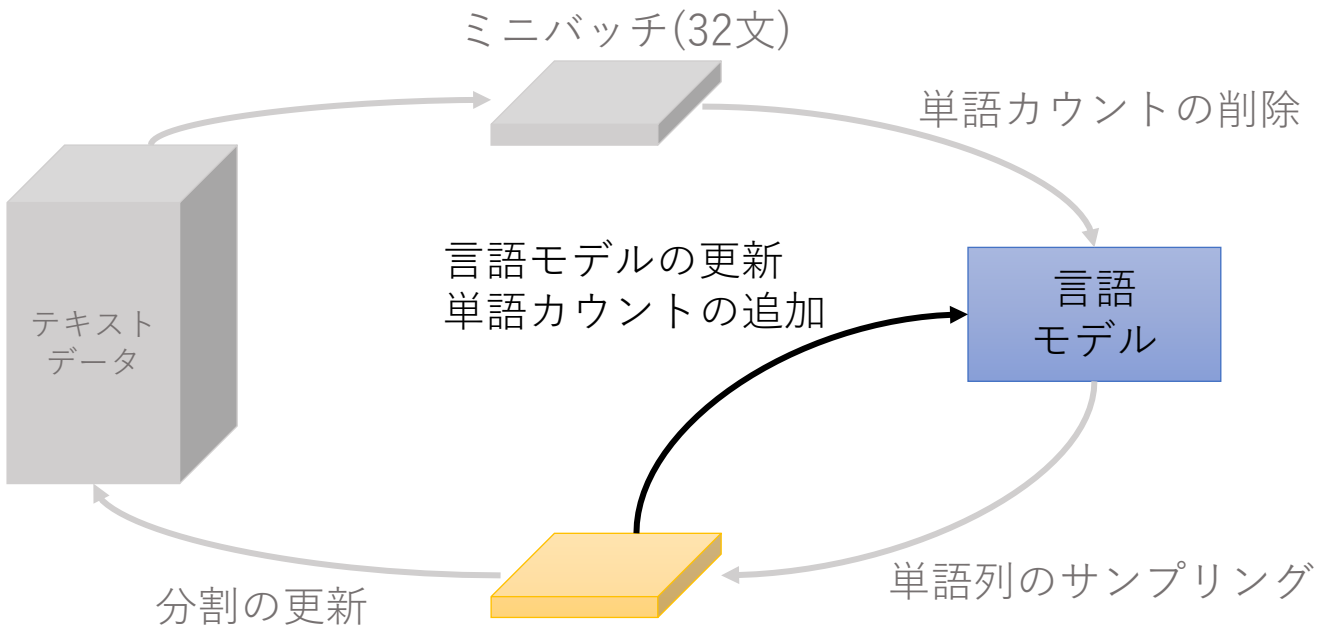
学習イテレーション

テキストをランダムに分割
```
for j <= maxEpoch:
  for each mini_batch:
```
ミニバッチに含まれる単語のカウントを削除

言語モデルから単語列をサンプリング

新たな単語列で言語モデルを更新
単語カウントの追加

現在の単語分割を保存



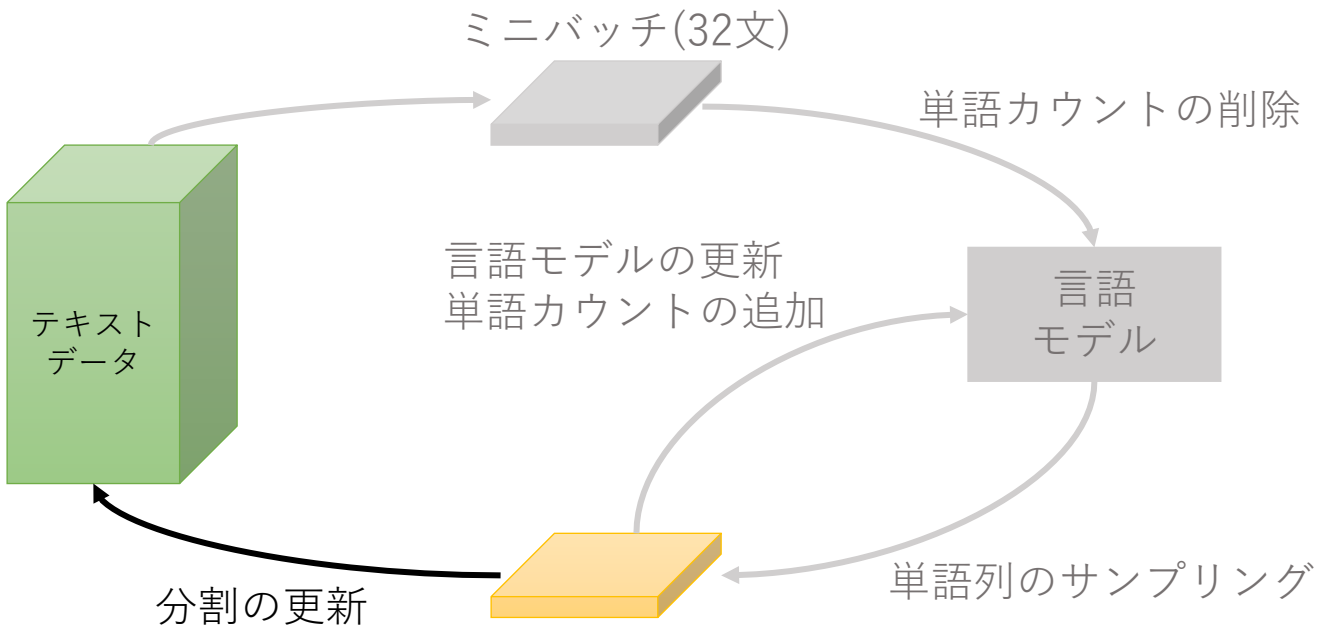
学習イテレーション

テキストをランダムに分割
```
for j <= maxEpoch:
  for each mini_batch:
```
ミニバッチに含まれる単語のカウントを削除

言語モデルから単語列をサンプリング

新たな単語列で言語モデルを更新
単語カウントの追加

現在の単語分割を保存



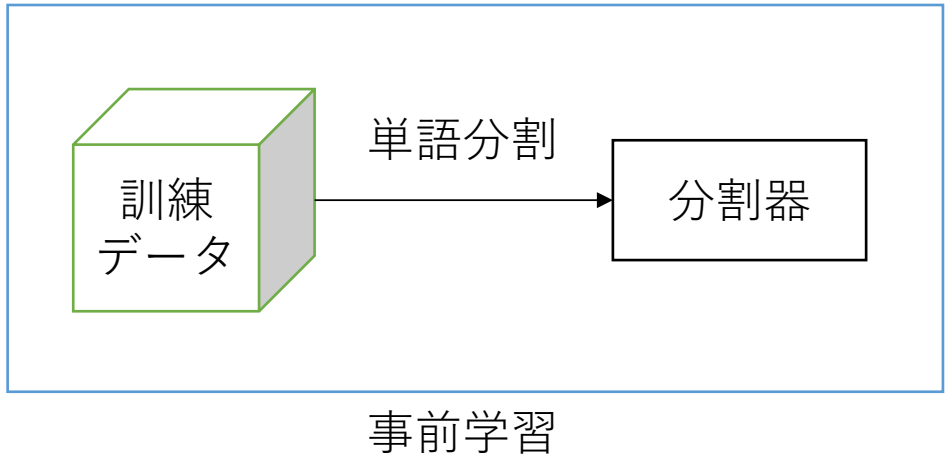
実験

- 教師なし単語分割の直接的な評価は難しい

- 得られた分割と分散表現を用いた文分類タスクで評価

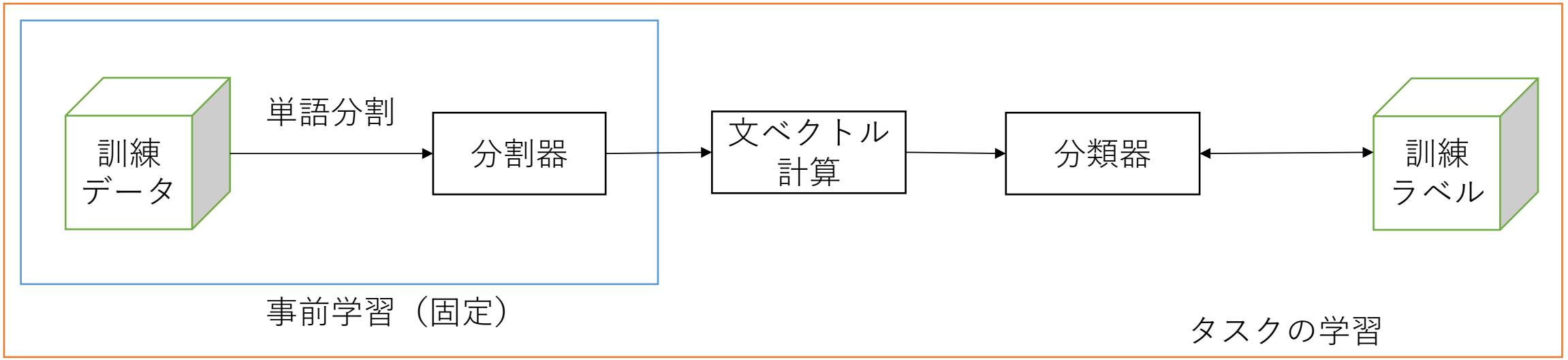
実験

- 提案手法を用いて訓練データに対して分割を学習
- 学習後のパラメータを用いてタスクを解く



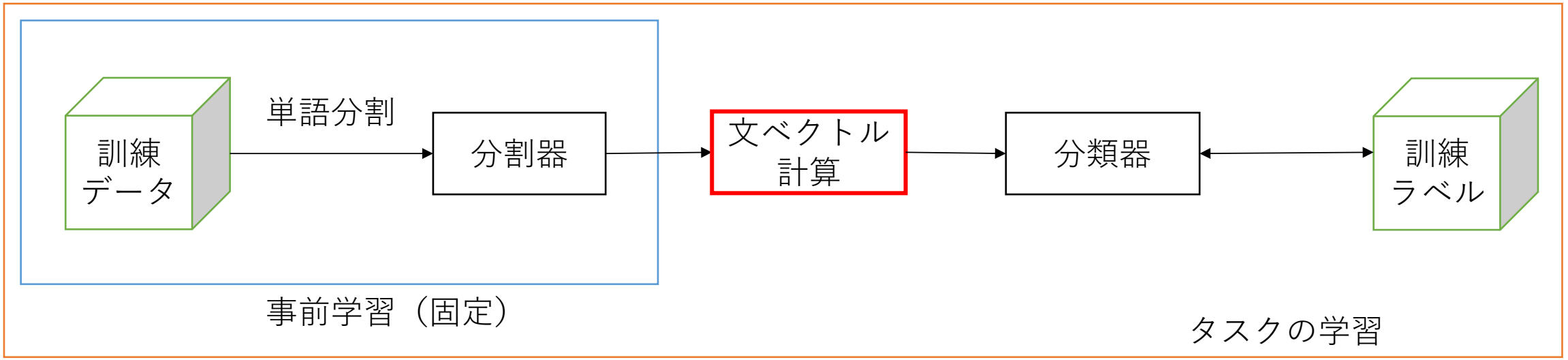
実験

- 提案手法を用いて訓練データに対して分割を学習
- 学習後のパラメータを用いてタスクを解く



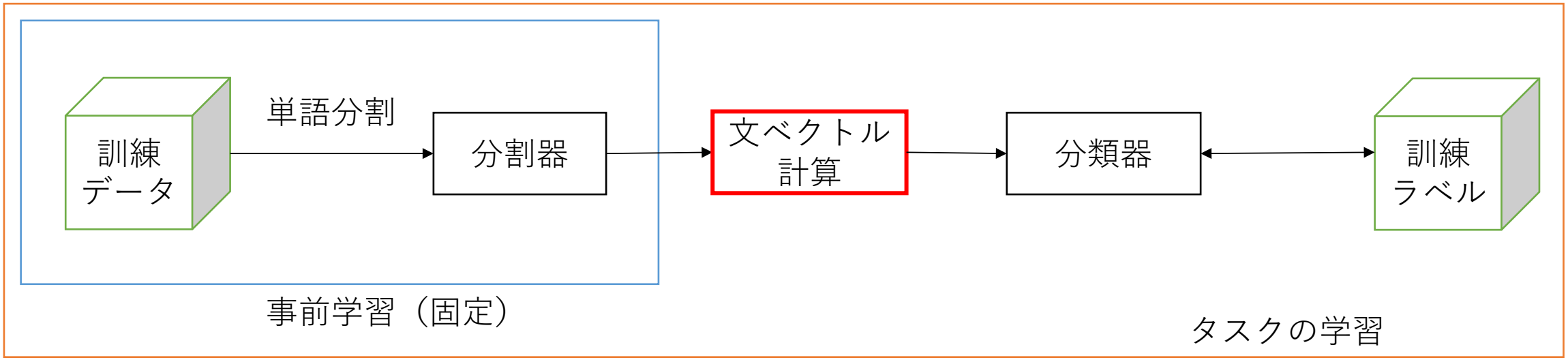
実験

- 文ベクトルの計算1
  - Long Short Term Memory (LSTM)
    - 系列を入力とするニューラル層



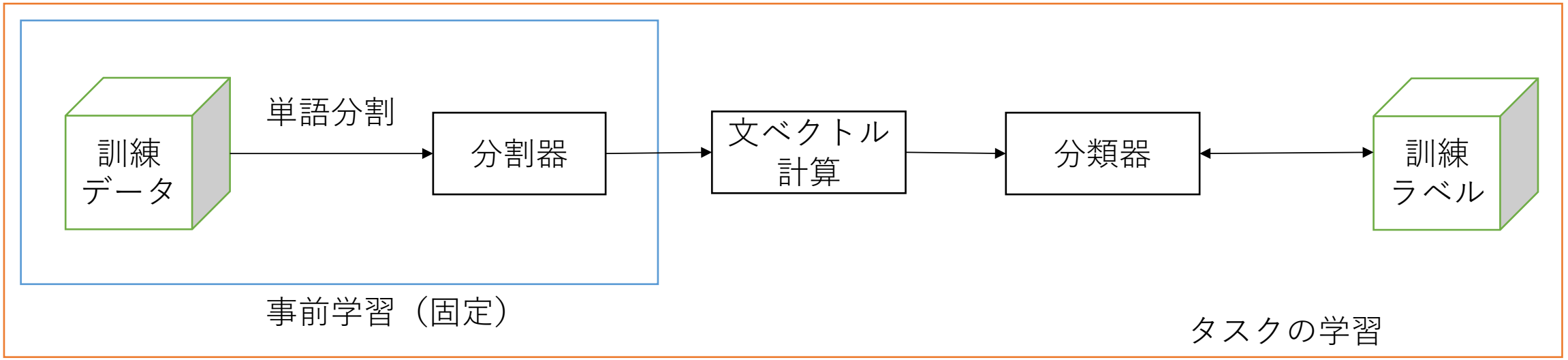
実験

- 文ベクトルの計算2
  - Deep Average Network (DAN)
    - 分散表現の平均を用いて文ベクトルを計算



実験

- 比較する分割手法
  - MeCab
  - Byte Pair Encoding (BPE)
- 単語レベルの言語モデルによって分散表現を事前学習



結果

表 3　分割による性能比較 (F 値)

	Twitter		NTCIR6	
	DAN	LSTM	DAN	LSTM
BPE	0.7044	**0.7057**	0.4818	0.4744
MeCab	0.7037	0.6978	0.4791	0.4673
提案手法	**0.7046**	0.7034	**0.4834**	**0.4830**
BPE+MeCab	0.7039	-	0.4833	-
BPE+提案手法	0.7042	-	0.4819	-
MeCab+提案手法	0.7040	-	**0.4866**	-
BPE+MeCab+提案手法	**0.7048**	-	0.4814	-

- データセット
  - Twitter評判分析データセット
  - NTCIR-6 感情分析パイロットタスク（毎日新聞）
- 考察
  - 提案手法による分割と分散表現を用いることで精度の改善が見られる
  - DANにおいては，提案手法による分割を既存手法の分割に足し合わせることで，精度の向上が見られる．

分割の考察

- 「もつやん」，「やばいな」といった，評判分析タスクを解く手がかりとなる部分について適切な分割が得られた

表 3　分割による性能比較 (F 値)

	Twitter		NTCIR6	
	DAN	LSTM	DAN	LSTM
BPE	0.7044	**0.7057**	0.4818	0.4744
MeCab	0.7037	0.6978	0.4791	0.4673
提案手法	**0.7046**	0.7034	**0.4834**	**0.4830**
BPE+MeCab	0.7039	-	0.4833	-
BPE+提案手法	0.7042	-	0.4819	-
MeCab+提案手法	0.7040	-	**0.4866**	-
BPE+MeCab+提案手法	**0.7048**	-	0.4814	-

表 4　手法による分割比較

データ	手法	分割
Twitter(訓練)	BPE	半 | 日 | ぐらい | 図 | 書 | 館 | に | いて | お | も | た | 。 | iPhone6S | め | っ | さ | も | つ | やん
	MeCab	半日 | ぐらい | 図書 | 館 | に | いて | おもた | 。 | iPhone | 6 | S | め | っさ | も | つや | ん
	提案	半日 | ぐらい | 図 | 書館 | に | いて | お | もた | 。 | iPho | ne | 6S | めっ | さ | もつ | やん
Twitter(評価)	BPE	2 | 日 | 充電 | せん | でも | い | けた | iPhone6S | plus | やば | いな
	MeCab	2 | 日 | 充電 | せん | でも | いけた | iPhone | 6 | Splus | や | ばい | な
	提案	2 日 | 充電 | せん | でも | いけ | た | iPho | ne | 6S | plus | やば | いな

分割の考察

- 新聞データは語彙の難易度が高く低頻度語が多いため，特に評価データにおいて分割の失敗が見られた．

表 3　分割による性能比較 (F 値)

	Twitter		NTCIR6	
	DAN	LSTM	DAN	LSTM
BPE	0.7044	**0.7057**	0.4818	0.4744
MeCab	0.7037	0.6978	0.4791	0.4673
提案手法	**0.7046**	0.7034	**0.4834**	**0.4830**
BPE+MeCab	0.7039	-	0.4833	-
BPE+提案手法	0.7042	-	0.4819	-
MeCab+提案手法	0.7040	-	**0.4866**	-
BPE+MeCab+提案手法	**0.7048**	-	0.4814	-

表 4　手法による分割比較

データ	手法	分割
NTCIR6(訓練)	BPE	喫 | 煙 | 論 | 争 | が | なか | なか | 決 | 着 | しない | 四 | つ | の問題 | 点 | を | 指摘 | した | い。
	MeCab	喫煙 | 論争 | が | なかなか | 決着 | し | ない | 四つ | の | 問題 | 点 | を | 指摘 | し | たい | 。
	提案	喫煙 | 論争 | が | なかなか | 決 | 着 | しない | 四つ | の | 問題点 | を | 指摘 | したい | 。
NTCIR6(評価)	BPE	第 | 二 | に | 基礎 | 学力 | の | 重 | 視 | にも | 半 | 歩 | 前 | 進 | が | 見 | られ | る。
	MeCab	第 | 二 | に | 基礎 | 学力 | の | 重視 | に | も | 半 | 歩 | 前進 | が | 見 | られる | 。
	提案	第二 | に | 基礎 | 学 | 力の | 重視 | にも | 半 | 歩 | 前進 | が | 見られる | 。

実験：学習速度の評価

- Twitterデータセット9,998文に対する文あたりのサンプリング・更新速度

	秒/文	50 epoch到達時間
友利ら	11.56 秒/文	96,314分
提案	**0.28 秒/文**	**2,332分**

- 提案手法により大きく改善
  - 軽量な言語モデル，動的計画法，バッチ学習を用いることで実現
  - 現実的な時間内での計算が可能

課題

- さらなる高速化と分割精度の向上
  - 省略した確率計算を盛り込む
  - 1万文を超えるテキストに対しても現実的な時間で分割と分散表現が得られるように改善していく
- 後段タスクの情報を用いた分割および分散表現の獲得
- 他言語への応用
  - 中国語，英語など

まとめ

- ニューラル言語モデルを用いた教師なし単語分割についてモデルの提案を行った
  - 軽量なニューラル言語モデルを用い，現実的な時間で計算可能
- 得られた単語分散表現を用いた実験を行なった
  - 日本語文分類タスクに対し，一定の精度向上に寄与