平均情報量による歌詞生成モデルの評価手法

李睿涵　山口和紀
東京大学

発表者：李睿涵

目的：
歌詞の特徴による歌詞生成モデルの評価手法の改善

本研究の貢献：
歌詞における平均情報量関連の特徴を調べた
平均情報量に基づく指標を三つ作り，新たな評価手法を提案した
特定の生成モデルによら，歌詞を評価する事が可能となる

先行研究と問題点

主観評価：人間の主観評価を正解と考えるので，被験者を用いて，歌詞の文法，意味，流暢さなどの観点から採点し，その結果を分析する

問題点：評価された歌詞と評価する人間のサンプル数を増やすことが難しいので，サンプルが偏り，効率が悪いという問題がある

客観評価：Perplexity, BLEU, rhyme density, etc.

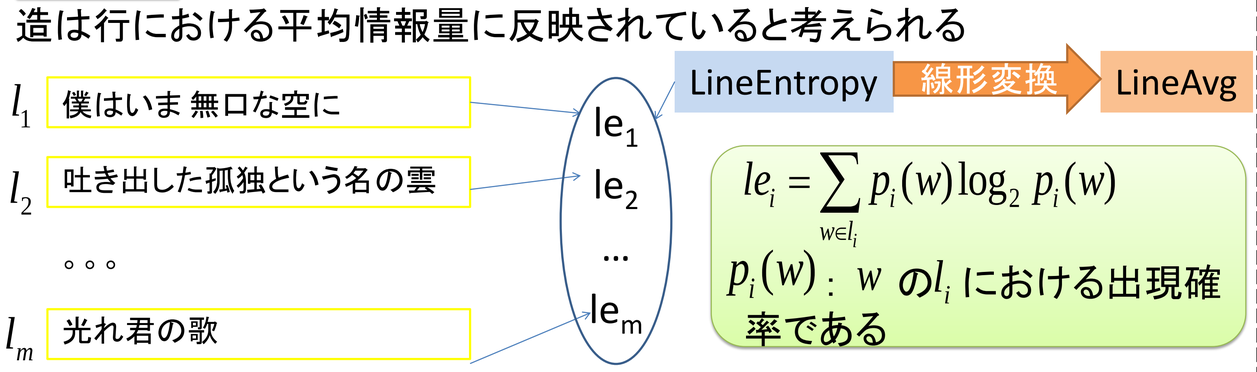
問題点：順序を考えず、文と文のつながりを評価できない；新しい歌詞を生成するときに正解がないので使えない；一般的な歌詞らしさを評価できない

提案手法

•歌詞文書はストーリーや感情などに関してすべて文字という媒体により表現するので，**単語の前後順序を考える**.
•文字一つずつの順序を評価することがかなり難しいので，歌詞の持つ**繰り返しの構造**に着目し，歌詞を特徴づける指標：
・行情報量の遷移
・累積行情報量の遷移
・累積行情報量増減の遷移
として平均情報量に基づくものを提案する.

行情報量の遷移(LineAvg)

Intuition：内容が同じ行の平均情報量は等しいので、歌詞の繰り返し構造は行における平均情報量に反映されていると考えられる

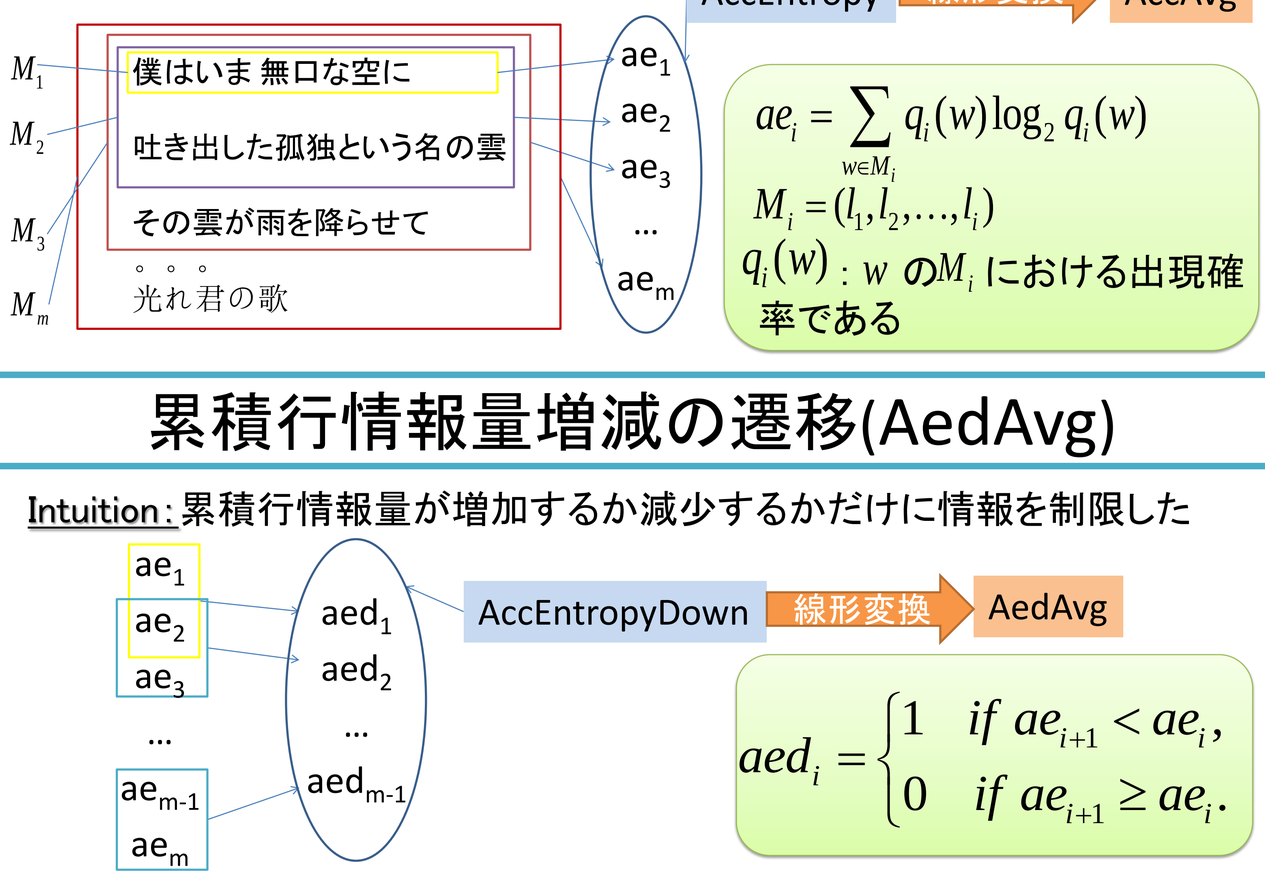


$$le_i = \sum_{w \in l_i} p_i(w) \log_2 p_i(w)$$

$p_i(w)$：w のl_i における出現確率である

累積行情報量の遷移(AccAvg)

Intuition：繰り返した行は既に出現した行とほぼ同じであるため、歌詞の先頭から繰り返した行まで見れば、得られる平均情報量はその前の行までの平均情報量よりも減る



$$ae_i = \sum_{w \in M_i} q_i(w) \log_2 q_i(w)$$

$M_i = (l_1, l_2, \ldots, l_i)$

$q_i(w)$：w のM_i における出現確率である

累積行情報量増減の遷移(AedAvg)

Intuition：累積行情報量が増加するか減少するかだけに情報を制限した

$$aed_i = \begin{cases} 1 & if\ ae_{i+1} < ae_i, \\ 0 & if\ ae_{i+1} \geq ae_i. \end{cases}$$

指標の次元の標準化

•実験で使用したデータごとに行数も異なる
•行数の違いにより歌詞と非歌詞が判別されるのを防ぐため、線形補間により指標のベクトルの次元を統一した

統一済みのベクトルの次元は**10**にした

0~10%　10%~20%　90%~100%

$$LineEntropy = (le_1, le_2, \ldots\ldots\ldots\ldots, le_m)$$

$$LineAvg = (line\_avg_1, line\_avg_2, \ldots, line\_avg_{10})$$

$AccEntropy \Rightarrow AccAvg = (acc\_avg_1, acc\_avg_2, \ldots, acc\_avg_{10})$

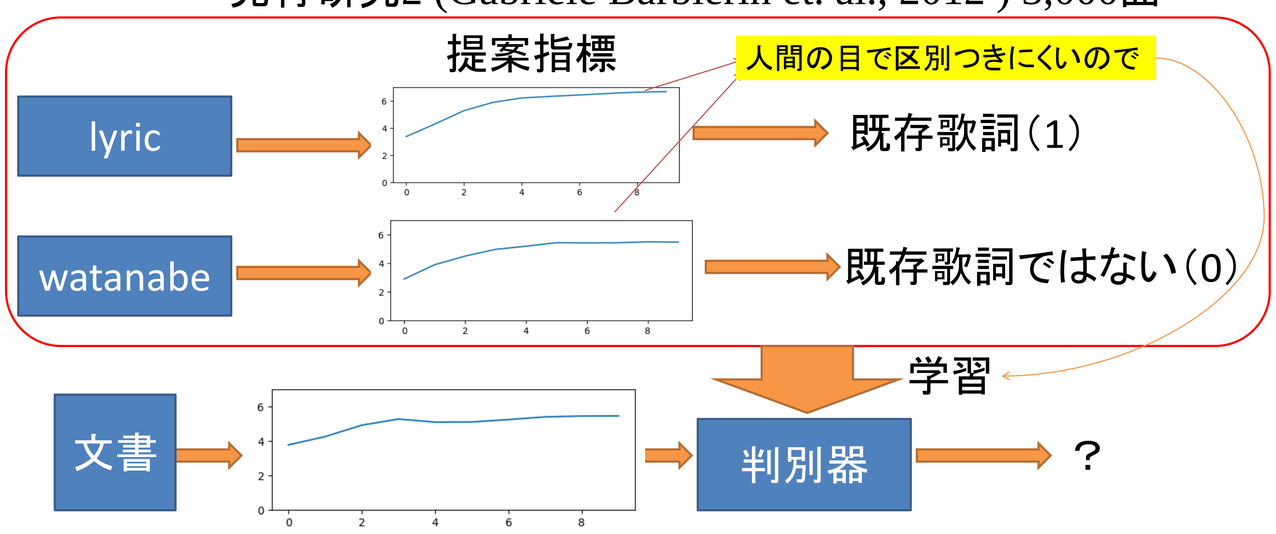
$AccEntropyDown \Rightarrow AedAvg = (aed\_avg_1, aed\_avg_2, \ldots, aed\_avg_{10})$

データ

歌詞データ → 既存の歌詞データ(lyric) 58,415曲

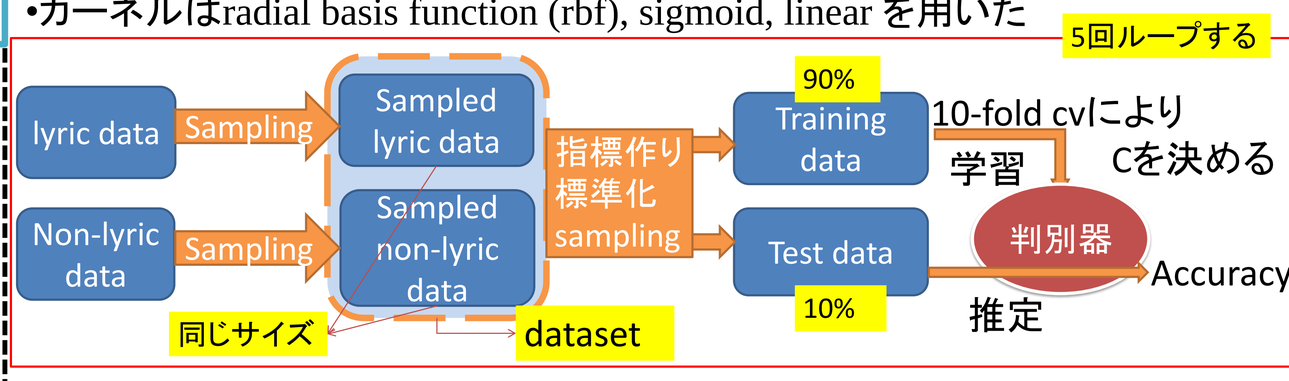
非歌詞データ：
既存の非歌詞データ：
ウェキペディア(wiki) 14,111文書
小説(aozora) 14,693冊

自動生成された歌詞データ：
先行研究1(Kento Watanabe et. al., 2014) (watanabe)100曲
先行研究2 (Gabriele Barbierin et. al., 2012) 3,000曲



実験設定

•トークン化：MeCab (http://taku910.github.io/mecab/)
•LineAvg, AccAvgとAedAvgを特徴量として用いて，SVMに学習させる
•カーネルはradial basis function (rbf), sigmoid, linear を用いた



モデルの判別性能としては5回サンプリングしたデータのテストデータによるAccuracyの平均を用いる

実験結果と結論

表 1: 歌詞データと非歌詞データの判別性能. 括弧内は各実験で用いたデータのサイズ (歌詞数, 文書数) である

	AedAvg			AccAvg			LineAvg		
	linear	rbf	sigmoid	linear	rbf	sigmoid	linear	rbf	sigmoid
lyric(5,000)+lyric(5,000)	0.502	0.502	0.501	0.513	0.493	0.497	0.503	0.505	0.495
lyric(3,000)+cm(3,000)	0.711	0.737	0.709	0.813	0.853	0.546	0.825	0.886	0.502
lyric(100)+watanabe(100)	0.750	0.740	0.750	0.980	0.980	0.570	0.960	0.970	0.020*
lyric(5,000)+wiki(5,000)	0.851	0.864	0.852	0.837	0.892	0.525	0.802	0.957	0.566
lyric(5,000)+aozora(5,000)	0.846	0.885	0.838	0.903	0.921	0.500	0.610	0.680	0.495

* 歌詞データが少なすぎて学習に失敗している. lyric(200)+watanabe(100) では 0.667, lyric(300)+watanabe(100) では 0.750 となる.

•歌詞と非歌詞とある程度判別することができた
•情報量の増減しか表しておらず、単語数の多さによる影響を緩和したAedAvgでも70%以上判別できた
•AedAvgの結果により、先行研究で生成された歌詞（cmとwatanabe）は非歌詞（wikiとaozora）より歌詞と判別しにくい
•行ごとの単語数による影響がより大きいAccAvgとLineAvgではAedAvgより判別性能が高い

今後の課題

•単語数と行数の影響を緩和し，評価手法を改善する
•本研究提案した指標を入れる歌詞生成モデルを作る