

# Transformer を用いた日本語併置型駄洒落の自動生成

畠山 和久

東京工業大学情報理工学院

hatakeyama.k.aa@m.titech.ac.jp

徳永 健伸

東京工業大学情報理工学院

take@c.titech.ac.jp

## 1 はじめに

雑談対話システムにおいて機械がユーモアを理解できるようになることで対話の継続性を高め、システムの性能向上が見込まれている [1]. 駄洒落は音韻的ユーモアを含む短文ユーモアとされており、広義では比喩の一種である [2]. たとえば「布団が吹っ飛んだ」において「布団」に音韻的に類似している「吹っ飛ん(だ)」という句が続いており、文の意味と文が含む音の観点での言葉遊びからこの文は駄洒落として成立していると考えられる. 駄洒落を自動生成できれば、機械がユーモアを理解するという点において有用である. 本研究では入力として与えた単語をお題とする日本語の駄洒落を自動生成することを目的とする.

## 2 関連研究

駄洒落は「かけられている部分 (種表現)」とそれに音韻的に類似する「かかっている部分 (変形表現)」の関係として捉えることができる. 種表現と変形表現が一文中に併置されているものを併置型、全体もしくは一部分を共有しているものを重畳型と呼ぶ. 重畳型では種表現は背景知識や文脈からくるものとされており、文中に明示されないことが多い. さらに併置型は、種表現と変形表現が音韻的に完全に一致する Perfect 型とそうでない Imperfect 型に細分化されている [3]. 表 1 に駄洒落の例を示す.

表 1 駄洒落の例

併置型 (Perfect)	医師の石
併置型 (Imperfect)	きちんと整頓されたキッチン
重畳型	このコーヒーは「あっ、高い」

荒木らはユーザー投稿型の 9 つの駄洒落サイトから収集した 68,000 件のデータに駄洒落の種別と種表現/変形表現区間をアノテーションし、駄洒落データベースを構築している [4]. 駄洒落を生成する試みとしてキムらはテンプレートとそれを埋める

スキーマを定義し、辞書と類音語リストを基に日本語の駄洒落なぞなぞの生成を行った [5]. 荒木はお題からクエリを生成し、駄洒落データベース内を検索する駄洒落生成システムを提案した [6]. Yu らは LSTM による Seq2Seq を用いて、初めて駄洒落を機械学習の枠組みで自動生成する手法を提案した [7]. 英語の重畳型駄洒落において、表層が一致する単語を 2 つの語義として解釈できるように並列に復号化することで駄洒落を行っている. また、駄洒落を認識するタスクについて、谷津らは語彙素性や子音列の音韻類似性を基にした SVM を用いる手法を提案した [8]. 本研究では機械学習の手法を用いて日本語を対象とした駄洒落の自動生成を行う.

## 3 提案手法

本研究では、Transformer [9] を用いた言語モデルを使って駄洒落を生成する. 言語モデルとして名大会話コーパス [10] から作成した Base Language Model (BLM) と荒木らの駄洒落データベース [4] から作成した Repetition Language Model (RLM) を考え、さらに、両者をマルチタスク学習して得られたモデル (MLM) によって駄洒落を生成する.

### 3.1 BLM

BLM は Encoder-Decoder モデルを用いて、お題とお題を含む一文を対応させて学習する. BLM を導入することによって、既存の駄洒落にない言い回しを生成することと駄洒落の流暢さを担保すること狙っている.

### 3.2 RLM

RLM は駄洒落における種表現と変形表現に見られる音の反復を学習することを狙っている. 文中の 2 箇所の区間が一定以上の類似度を持てば駄洒落だと考えられるが、駄洒落では種表現と変形表現の位置が自明ではなく、一般的な (駄洒落でない) 文であっても短い区間が偶然類似する可能性もあ

る。そこで本研究では編集距離(レーベンシュタイン距離) [11] を参考にして設計した, 2つの文字列の類似度を計算する関数  $S_{\text{pun}}$  を用いて, 文字配列  $S_q$  について音の反復を定量化するスコア PRS (Pun Repetition Score) を以下のように定義する。

$$\text{PRS} = \max_{2 \leq i \leq |S_q| - 2} S_{\text{pun}}(S_q[0 : i], S_q[i : |S_q|])$$

$$S_{\text{pun}}(X, Y) = \max_{0 \leq i \leq |X|, 0 \leq j \leq |Y|} \frac{\text{DP}[i, j]}{\min(i, j)}$$

ただし,

$$\text{DP}[i, 0] = \text{DP}[0, j] = 0$$

$$\text{DP}[i, j] = \max \begin{cases} \text{DP}[i - 1, j] + \text{Skip}[i - 1, j] & (1), \\ \text{DP}[i - 1, j - 1] + \text{sim}(X[i], Y[j]) & (2), \\ \text{DP}[i, j - 1] + \text{Skip}[i, j - 1] & (3) \end{cases}$$

$$\text{Skip}[i, j] = \begin{cases} 0 & (\text{DP}[i, j] \text{ が (2) 式より得られた}) \\ -0.8 & (\text{otherwise}) \end{cases}$$

$$\text{sim}(x, y) = \text{match}(x_v, y_v) + \text{match}(x_c, y_c)$$

$$\text{match}(x, y) = \begin{cases} 1 & (x = y) \\ -0.2 & (x \neq y) \end{cases}$$

DP, Skip は  $S_{\text{pun}}$  を計算する際に使う二次元配列,  $x_v, x_c$  はそれぞれ文字  $x$  の母音, 子音を表す。  $S_{\text{pun}}$  は編集距離に以下の変更を加えている。

- 入力となる文字列  $S_q$  には読み仮名列を考える。各文字の母音と子音も考慮する。
- 編集距離が不一致部分での挿入/削除/置換に対して減点されるのに対して  $S_{\text{pun}}$  は類似部分について加点する
- $S_{\text{pun}}$  の挿入/削除は配列 Skip を使い, 2文字以上マッチしない場合に減点する。
- 一致もしくは置換のコストは母音と子音それぞれについて  $\text{match}(x, y)$  で計算する。

PRS によって音の重複の観点から文字列が駄洒落らしいかどうかを評価できる。

音の反復を明示的に生成する手法として, 通常の言語モデル学習で用いられる Cross Entropy 損失 ( $L_{\text{CE}}$ ) のみのモデルに加え, お題の読み仮名を入力に加えた上で以下の KL Divergence 損失 ( $L_{\text{KL}}$ ) を加えるモデルも考慮する。<sup>1)</sup>

$$\text{Loss} = L_{\text{CE}}(y, t) + \min_i L_{\text{KL}}(y[i : i + |x'|], D(x')) \quad (4)$$

ここで  $x'$  は入力の単語のひらがな表記,  $y$  はニューラルネットワークにおける出力の確率分布,

1) 入力として与えられる単語は種表現として解釈する

$i$  は教師信号,  $D$  は文字を要素とする語彙  $V$  に対して  $|V| \times |V|$  の二次元配列で,  $D(\cdot)$  は引数に含まれる文字に対応する列を切り出し結合した  $|V| \times |x'|$  の部分配列を表わす。  $i$  はモデルの出力を読み仮名の長さ分切り出す際の開始位置を表している。  $L_{\text{KL}}$  では出力の系列に対して一部分を切り出しながら入力単語の読み仮名との類似度を計算し, 尤もらしい部分に対して類似度  $S_{\text{pun}}$  による損失を与える。この損失を加えることにより出力の一部分に入力の読み仮名と似た部分を生成しようとしている。

$L_{\text{KL}}$  を計算するには教師分布が必要であるが, 式 (4) 中の  $L_{\text{KL}}$  の第2引数  $D$  として  $V$  中の各文字に対し, 自分自身のみを1, それ以外を0とする二次元配列  $U$  と, 各文字に対して  $V$  中のすべての文字との類似度を  $S_{\text{pun}}$  によって計算した二次元配列  $W$  の2種類を設定した。以降この配列  $U, W$  を教師分布と呼ぶ。  $U, W$  は以下のように定義する。

$$U_{ij} = \begin{cases} 1 & (i = j) \\ 0 & (\text{otherwise}) \end{cases}$$

$$W_{ij} = S_{\text{pun}}(v_i, v_j) (v_i, v_j \in V)$$

### 3.3 MLM

BLM と RLM を基にマルチタスク学習を行う。駄洒落データベース [4] と名大会話コーパス [10] 両方を使用し, それぞれタスクで用いた損失関数の重み付け和で学習を行う。

## 4 PRS の評価

駄洒落生成に先立って, 駄洒落識別タスクを用いて PRS の妥当性を確認する。

### 4.1 データ

正例として駄洒落データベース [4] を, 負例として名大会話コーパス [10] を用いる。駄洒落データベースの前処理として各種アノテーションを削除し, 生データを再現した形態にした。名大会話コーパスの前処理として句点や「!」, 「?」で終わっているものを一文として扱う分割を行った。また, 「ぬるような, 冷たいような...」は駄洒落ではないが「ような」の区間で反復が起こっている。このような駄洒落を意図していない反復表現に対処するため, 既に一致している区間を削除する。

## 4.2 実験

ベースラインとして谷津ら [8] の手法を用いる。谷津らは学習セットの語彙を用いた bag of words と読み仮名列の一致素性, 子音の音韻類似度素性等を基に SVM で分類器を作成している。PRS では得られたスコアの閾値を学習セットを用いて 0.75 とした。正例, 負例それぞれ 250 件, 計 500 件のデータを 5 分割交差検証して得られた Precision, Recall, F1 値の平均を表 2 に示す。PRS による方法は F1 値ではわずかにベースラインに及ばないものの, Precision に関しては SVM に比べ高い性能を得ていることから, PRS を駄洒落の音韻的な観点での評価に用いる。

表 2 PRS 実験結果

手法	Precision	Recall	F1
SVM	0.816	0.956	0.880
PRS	0.904	0.824	0.862

## 5 駄洒落生成の評価

### 5.1 データと前処理

BLM の学習には名大会話コーパス [10] を, RLM の学習に駄洒落データベース [4] を用いる。それぞれお題となる入力 1 単語から 1 文を生成するタスクを学習する。名大会話コーパスでは以下の前処理を行った。

1. 会話文を句点で分割し, 基本的に一文を 1 単位として扱う。
2. 30 文字以上の文を読点で分割しそれぞれを 1 単位に変更する。4 文字以下の文章は削除する。
3. MeCab<sup>2)</sup> で形態素解析を行い, 名詞を抽出しお題と定義する。出力文のお題部分は [MSK] トークンに置き換える。

駄洒落データベースでは, 併置型駄洒落のみを対象として, 種表現をお題とし, 駄洒落のお題部分を [MSK] トークンで置換する前処理を行った。それぞれのデータについてお題を基にグループ化し, お題によって学習/テストデータの分割を行った。これらの処理により, 名大会話コーパスから 157,132 文, 駄洒落データベースから 52,246 文の学習データが得られた。

2) <https://taku910.github.io/mecab/>

### 5.2 モデル

BLM, RLM, MLM 全てにおいて, エンコーダー, デコーダーそれぞれ 12 層の Transformer を用いて実験を行った [12]。また, お題の単語を確実に出力するため, コピー機構 [13] を参考に [MSK] タグを入力単語に置き換える。実験設定として, epoch 数 = 40, optimizer=AdamW, 学習率 =  $2 \times 10^{-5}$ , drop out = 0.1, MLM における重みは  $\alpha = \beta = 0.5$  で実験を行った。tokenizer は形態素でなく文字単位の分割を行っている。

### 5.3 自動評価

自動評価では BLM,  $L_{CE}$  のみを用いた RLM(以下 RLM-CE), 教師分布  $U$  を用いた  $L_{KL}$  を損失関数に追加した RLM(以下 RLM- $KL_U$ ), 分布  $W$  を用いた  $L_{KL}$  を追加した RLM(以下 RLM- $KL_W$ ), Cross Entropy 損失のみを用いた MLM(以下 MLM-CE), 分布  $U$  を用いた  $L_{KL}$  を追加した MLM(以下 MLM- $KL_U$ ), 分布  $W$  を用いた  $L_{KL}$  を追加した MLM(以下 MLM- $KL_W$ ) の 7 つのモデルを比較する。評価項目はパープレキシティ (PPL), 異なり 1-gram, 異なり 2-gram, PRS を用いる。結果を表 3 に示す。

表 3 自動評価の結果

Model	PPL	1-gram	2-gram	PRS
BLM	<b>8.86</b>	0.027	0.200	0.656
RLM-CE	99.2	0.045	0.320	0.856
RLM- $KL_U$	68.5	0.067	0.390	<b>0.869</b>
RLM- $KL_W$	80.6	<b>0.081</b>	<b>0.437</b>	0.857
MLM-CE	72.5	0.065	0.385	0.793
MLM- $KL_U$	49.2	0.052	0.329	0.844
MLM- $KL_W$	75.0	0.064	0.379	0.803

パープレキシティは言語モデルの学習における確信度を表し, 文章の形態が独特である駄洒落のデータを用いることで値が大きくなったと考えられる。異なり 1-gram, 異なり 2-gram では全てのモデルについて文字単位の分割を考えていることから低い数字になっている。PRS の評価では 4 節で採用した 0.75 を閾値としている。BLM 以外のモデルは平均的に見ると音韻的には駄洒落を生成しているといえる。

### 5.4 人手評価

RLM-CE, MLM-CE, MLM- $KL_U$ , MLM- $KL_W$ , 荒木の生成手法 [6] の 5 つのモデルの出力に対して人

表4 可読性, 流暢性, 面白さの評価 (平均)

Model	可読性	流暢性	面白さ
荒木ら	<b>3.98</b>	<b>4.11</b>	<b>3.07</b>
RLM-CE	2.65	2.87	2.03
MLM-CE	2.54	2.84	1.84
MLM-KL <sub>U</sub>	2.44	2.65	1.90
MLM-KL <sub>W</sub>	2.42	2.70	1.84

表5 妥当性, 作者の評価

Model	妥当性		作者		
	妥当	不適	機械	人間	不明
荒木ら	89/719	11/281	20/241	71/552	9/207
RLM-CE	39/379	61/621	79/566	15/210	6/224
MLM-CE	20/238	80/762	88/647	8/152	4/201
MLM-KL <sub>U</sub>	22/295	78/705	86/627	8/140	6/233
MLM-KL <sub>W</sub>	17/249	83/751	84/631	10/128	6/241

手評価を行った。入力となるお題は分類語彙表 [14] の見出し語から 100 語を選択した。

なお, 荒木の手法において一部のお題に対しては生成ができないので, この評価では荒木の手法で生成が可能な見出し語 100 語を用いている。

評価項目は可読性 (理解できるか), 流暢性 (文法的か), 面白さ, 駄洒落としての妥当性, 作者 (人間が機械か) の 5 項目である。可読性, 流暢性, 面白さに関しては 1~5 までの 5 段階で, 妥当性については二値 (妥当/不適), 作者については三値 (人間/機械/不明) で評価する。ラテン方格法を用いて 100 のお題に対し各モデルの出力が 20 個ずつ含まれるように 5 種類の評価シートを作成し, クラウドソーシングを用いて各評価シートにつき 10 名に評価を依頼した。可読性, 流暢性, 面白さの平均値を表 4 に, 妥当性, 作者の結果を表 5 に示す。妥当性と作者については各出力の 10 名の判定の多数決をとって数えた数字を左に, 個々の判定を単純に総和した数を右にスラッシュで区切って示した。

すべての評価項目において荒木らのモデルが一番数字は良いが, 荒木らの方法は人間が作った駄洒落を検索していることと, そもそも必ず検索できるお題しか入力して与えていないことから荒木らの結果はこの評価における上限と考えられる。今回と同じように分類語彙表からランダムに選択した見出し語をお題として与えた場合, 荒木らの方法では 70.8% に対してしか駄洒落を出力できないことが報告されている [6]。これに対して本研究では任意の入力に対し出力を生成することはできる。そのうち 2 割から 3 割程度は駄洒落として評価されることがわかっ

た。その中には人間が作ったと思わせる駄洒落も生成できている。

## 5.5 事例分析

お題「果報」に対する各モデルの出力例を表 6 に示す。

表6 お題「果報」に対する出力

モデル	出力
荒木ら	果報は寝て待てだ〜! キャッホー♪
RLM-CE	果報を買うほど!
MLM-CE	果報が見つかって快調!
MLM-KL <sub>U</sub>	果報を見て, かほーとしてたのか?
MLM-KL <sub>W</sub>	果報がかえるかどうかは, 火葬してられないか!

「果報」というお題に対して, RLM に比べて MLM では文長が長くなっており, より一文の文章らしい言い回しになっている。RLM のように「ほど!」という文末は一般的な文章で見られない駄洒落らしい言い回しであり, MLM には名大会話コーパスによる会話文のスタイルが反映されていると考えられる。MLM-KL<sub>U</sub> には果報の読み仮名「かほう」が文内にあり, 導入した損失と分布を反映した結果になっている。MLM-KL<sub>W</sub> では変形表現として「火葬」が含まれ, 読み仮名を元にお題の変形表現を生成できている。

## 6 おわりに

本研究では日本語駄洒落を機械的に自動生成することを目標とし, Transformer を用いた言語モデルの設計を提案した。名大会話コーパスによる一般的な文章生成タスクを併用するマルチタスク学習と変形表現を生成する損失関数により, 文章スタイルの異なる駄洒落生成を行った。

今後の課題として, 生成モデルとしての性能向上が上げられる。短文生成のタスクとして入力の情報量を増やしたり, 制約を加えることなどにより文章生成を制御する必要があると考えられる。また, 駄洒落はユーモアであるが本研究では面白さについては考慮をしていない。生成する変形表現や文のスタイル等に面白さの観点を導入することで, 面白く, 目新しい駄洒落の生成ができると考えられる。

## 参考文献

- [1] P. Dybala and K. Araki. Multiagent system for joke generation: Humor and emotions combined in human-agent conversation. *Journal of Ambient Intelligence and Smart Environments*, Vol. 2, No. 1, pp. 31–48, 2010.
- [2] 滝澤修. 記述された「併置型駄洒落」の音素上の性質. *自然言語処理*, Vol. 2, No. 2, pp. 3–22, 1995.
- [3] 佐山公一 谷津元樹 荒木健治. 駄洒落データベースの構築及び分析. *人工知能学会 第2種研究会 ことば工学研究会資料*, Vol. 2, No. 3, pp. 13–24, 2017.
- [4] 荒木健治, 佐山公一, 内田ゆず, 谷津元樹. 駄洒落データベースの拡張及び分析. *人工知能学会 第2種研究会 ことば工学研究会資料*, Vol. 3, No. 1, pp. 1–15, 2018.
- [5] キム・ビンステッド, 滝澤修. 日本語駄洒落なぞなぞ生成システム「boke」. *人工知能学会誌*, Vol. 13, No. 6, pp. 920–927, 1998.
- [6] 荒木健治. 駄洒落データベースを用いた駄洒落生成システムの性能評価. *人工知能学会 第2種研究会 ことば工学研究会資料*, Vol. 3, No. 8, pp. 39–48, 2018.
- [7] J. Tan Yu and X. Wan. A neural approach to pun generation. *In Association for Computational Linguistics (ACL)*, Vol. 1, pp. 1650–1660, 2018.
- [8] 谷津元樹, 荒木健治. 子音の音韻類似性及び svm を用いた駄洒落検出手法. *知能と情報 (日本知能情報フレンジ学会誌)*, Vol. 28, No. 5, pp. 833–844, 2016.
- [9] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N Gomez Lukasz Kaiser Ashish Vaswani, Noam Shazeer and Illia Polosukhin. Attention is all you need. *In Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.
- [10] 藤村逸子・大曾美恵子・大島ディヴィッド義和. 会話コーパスの構築によるコミュニケーション研究. 藤村逸子, 滝沢直宏編『言語研究の技法：データの収集と分析』ひつじ書房, pp. 43–72, 2011.
- [11] 編集距離 - 東京工芸大学, アクセス：2021/1/11. <http://www.cs.t-kougei.ac.jp/SSys/LevenS.htm>.
- [12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics.
- [13] Hang Li Jiatao Gu, Zhengdong Lu and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *In Association for Computational Linguistics.*, 2016.
- [14] 国立国語研究所 (2004) 『分類語彙表増補改訂版データベース』 (ver.1.0).

## 付録：RLM-KL<sub>U</sub> で生成された駄洒落の例

先攻のせいで先公センコーかいじゃない!  
新譜はシンプルなドレスを着ている!  
ハッチの件はちょっとやいたよと、はっち来なさい  
酒屋がさかやっかいになった!  
ノベルについて述べる!  
触手を扱うくしゅくれる人は、ショックです!  
すなわち、すなはちだけて食べない!  
襟については、ええエリしたことだ!  
むかつく婿だから、むう、勝つく!  
総括があるそうだよ。そつことを言ってる!  
もちろん、もちろんと食べよう!  
家のふきんに、フキンはないか?  
市民のしみんもんだ!  
いえという言葉は、いい絵いている  
彼らのカレーは辛かれられない!  
地方のちほうがいいんだって?そうはい、チホントですね。  
景色をけしき声が高いとは、意気をつけます!  
資源がしげーんない!  
おかしい形のかしはおおかしいかいしい  
マジシャンでまじ、やん。  
書棚をしょうだな  
一周にいてもいいしゅー  
コントロールで、コントロール!  
不完全のフカンぜん  
潮騒をしおおいしつけれ!  
装甲車をそう、うしゃん持ってるの?  
和書、わし、よく食べるよ!  
落としぶた夫おとしぶた。  
つてについては、うつて立たないこと!  
不屈で拭くつーの  
親権、しんけんはいつも、シンケンゲンをする!  
マグマを食べてから、顔がまぐまった!  
右折がうせつくのは、うー、初めて!  
リズムがいるのは、リーズム!  
のどのどが悪くてもわかります!  
塗りを作ったのは、ハリー・ガール  
速い、はあ、やい、どこにやって来て!  
占拠のせんきょうだ?  
軍手のぐんって、おならいだらう!  
新月なんてげつやがうまくないのは、シンゲつーのこ  
と言ってな  
スーパーマンの排ガスを吸うパーマン!  
副産物の服さん、ぶつかってるね!  
乳児がゆうじようになってる  
取り立てが取り立てた  
アルバムの上にあるバンド・アルバイト!  
激高の下で、かけっこーりした人がいる  
隔離の絵を描くリア  
執念深い、しゅうねえぶかい?  
暇人にひまじんないさいには、ひいー、マジンをたれま  
んじゃよ  
信徒が使う空港は、新東京国際空港!  
とわはとわかった  
背もたれのせいで、セモタレの下に埋めました  
お湯になったのは、おっゆーもんだ  
発行したのはうちに、8個はっこうしてられない

雲海を見に行いたのは、うん、かいーん  
雨季がうきるしじゃないから、よう気づいてる!  
応諾してうだくならオーダクリダック  
逆転ギャくてんの?  
そこはかとなく、そこはかとなくくない。  
知るの下に、何しる?  
新刊を震新刊かんに見つけた!  
圏外にはんけんがいらない  
長椅子の長いっす!  
アバンギャルドをあばんぎゃやると  
閉館と言われて、へい、感じさせてました  
抗菌にこう気をつける!  
隅が無くて済みました!  
マイコン買うなら、まあ、いい子の下に!  
うみのうみを使うのはウミングアップ  
湿田についての失点!  
ユーザー、ゆーざーない!  
長身は超ようしんがない!  
おいとまもらったのはおとまだ来いにおけません。  
ブランク、ぶらんくく?  
真紅がしんくじゃいけない!  
深雪に行くのは、おみゆきだあー  
こだまがこだまにある!  
賞金にはショウキングしょうきん持ちではありません  
聖者がせいじゃったのは、そんなに成人や!  
「いよいよ」という言葉は、「いよいよヨープレット」  
来聴はらいちょうぶ?  
とっくに、とっくにいー!  
果報をかほう?  
なぎになぎって見つからない  
弱音のようはねを食べている  
略称をりゃ、しょうがない  
うそぶく、うそぶくなってきたよ!  
法曹できるようになるのは、ほう、そうだね!  
支障ししょうがないなんて、うそうですか?  
ハロゲンでは、げんな所にハローゲン!  
無地になったら、身を蒸じっとしていた  
脱腸は、つちょうぶーがないねえー!  
船倉があるぐらいで、フナグラフグラ食べる  
着替えなきがえーないなんて、気が絵にしてる!  
協力のきょうりよくに教えてくれると言われて今日  
ショックエス  
進歩でしんぼを書いたらしい事が言いました!  
商家を紹介してくれたのは、師匠か!  
今日のおかしいしおらしい、誰でも知らんらしい。  
ANA ホールで、穴掘る。  
スタイリストのすたいリスト!