Comparison of Pun Detection Methods Using Japanese Pun Corpus

Motoki Yatsu* **and Kenji Araki****
*Aoyama Gakuin University,
5-10-1 Fuchinobe, Chuo-ku, Sagamihara, Kanagawa, 252-5258, Japan
yatsu@it.aoyama.ac.jp
**Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido, 060-0814, Japan
araki@ist.hokudai.ac.jp

Abstract
A sampling survey of typology and component ratio analysis in Japanese puns revealed that the type of Japanese pun that had the largest proportion was a pun type with two sound sequences, whose consonants are phonetically close to each other in the same sentence which includes the pun. Based on this finding, we constructed rules to detect pairs of phonetically similar sequences as features for a supervised machine learning classifier. Using these features in addition to Bag-of-Words features, an evaluation experiment confirmed the effectiveness of adding the rule-based features to the baseline.

Keywords: corpus, Japanese, humour, puns, detection

1. Introduction

In recent years, there has been growing interest in machines expressing and understanding humour. There is an interest in computational humour improving the quality of life (QOL) of the user, such as alleviating the psychological stress of the person expressing himself (Yamada et al., 2012), and improving depressive symptoms (Tsukawaki et al., 2011).

Research to obtain such effects in artificial agent and robot communication is under way. For example, Dybala et al. (2012) showed that the user feels unexpected to the system and the desire to continue dialogue with the system is increased under the condition that the dialog agent generates humour. Miyazawa et al. (2012) conducted a survey of factors that enhance the continuity of the dialogue between the dialogue system and the user. As a result, it was shown that a humourous dialogue would get enhanced its continuity.

In this paper, first we take a look at what types of Japanese pun are prominent from a categorization survey. Subsequently, we describe the rule-based and machine learning-oriented features as training data for a supervised machine learning method, reflecting the phonological characteristics that Japanese puns share.

(Seed) *Futon-ga futton-da.* (Stc. 1)
(Transformed)
The *futon* blew away.

Kiri-no naka-de kiri-o kiri taoshi-te mo
kiri-ga nai. (Stc. 2)
Even if you cut down paulownia in the fog, you will not reach the end.

Suimasenbazuru! (Stc. 3)
suimasen *senbazuru*
I apologize (with a thousand of paper cranes)!

Figure 1: Component ratios of the types of Japanese puns (*italic*) found in the pun corpus noted in Section 2.2.

2. Japanese Puns as Text Humour

2.1 Types of Japanese Puns

According to (Takizawa, 1995), Japanese puns can be classified mainly into two classes: **juxtaposed puns** and **superposed puns**.

In a **juxtaposed pun**, a **seed** expression refers to one or more independent (noun, verb, adjective, or emoticon) morpheme(s) or phrase(s) in the sentence. A **transformed** expression refers to a phoneme string in an arbitrary section in a sentence having phonological similarity with seed expression(s). For example, the first sentence (Stc. 1) shown in Figure 1 have a pair of phoneme sequences phonologically similar, one of which has an extra 't' character. As another example, Stc. 2 has more than one pair of similar phoneme sequences. These kinds of puns are classified into juxtaposed puns.

Kawahara and Shinohara (2012) define **perfect** puns as juxtaposed puns whose seed and transformed expressions have no phonological difference (as seen in stc. 2). Similarly, those that do not match due to changes or dropout of phonemes (stc. 1) are referred to as **imperfect** puns.

Superposed puns, on the other hand, have only one surface part with two or more semantic interpretation. An example is shown as Stc. 3.

2.2 Pun Database

We created a pun database, which consists of sentences as the source for Japanese juxtaposed pun humour texts, with a scale that has never been achieved. A web crawler was

Pun Type		Frequency
Juxtaposed puns	Perfect	42 (7.0%)
	Imperfect	521 (86.8%)
	Total	563 (93.8%)
Superposed puns		33 (5.5%)
Other		4 (0.7%)

Table 1: Component ratios of the types of puns in a sample of 600 sentences from pun corpus.

Website Name	URL	# of Sentences
Dajare Nabi	http://www.dajarenavi.net/pc/i_today_index.htm	39,120
Dajare Suteshon	http://dajare.jp	8,795
Dajare Netto	http://www.dajare.net	1,621
Hitokuchi Dajare Daishuugou	http://www.biwa.ne.jp/~aki-ina/gyagu.html	1,067
Dajare Shu Dajare Jiten	http://dajareshuu.web.fc2.com	982
Dajare No Kanzume	http://www.geocities.jp/pikumin_hiroba/dajare.html	572
Dajare Kurabu	http://with2.net/dajakura/	428
Dajare Hiroba	http://www1.ocn.ne.jp/~origo/dazyare	303
Dajare o Itta no ha Dareja?	http://wtpage.info/dajare/	107
Total number of extracted pun sentences		52,995
Number of pun sentences after removing dupicates & near-duplicates		45,970

Table 2 : The source websites from which pun sentences were obtained.

run to gather pun sentences from specified websites (shown in Table 2) that contained humourous sentences. Over 95% of the data obtained here is shared with the Japanese pun database created by Araki et al. (2017).

We obtained Japanese puns as positive data of a classification problem. Originally, in analyzing puns, as with many other linguistic phenomena, it is necessary to analyze them reflecting the frequency of occurrence in the real world and conditions for occurrence (such as topics with high possibility of occurrence of conversation including puns). However, with regard to the use of puns in everyday conversation, actual corpus and its statistical analysis result are not available so far and it is still difficult to find concrete and practical examples.

Therefore, we obtained the positive data from the Web using a spider. Duplicate and near-duplicates (with the difference of 3 characters or less) were removed at the time of acquisition. Table 2 shows the websites from which the data was obtained, the total number of extracted sentences, and the final number of obtained pun sentences. From the results of sampling survey we made against this pun corpus (shown in Table 1), we can see that Juxtaposed and imperfect puns have a large portion (> 85.0%).

3. Pun Detection Methods

In this research, we refer juxtaposed puns as the main object of detection. The reason is that, firstly, the number of sentences with juxtaposed puns has a large proportion of all the pun sentences, as we noted shortly in Section 2.1. We use a Support Vector Machine (SVM) as a supervised machine learning method, as a two-class classifier using vector space model with input as vector. We used *scikit-learn* [1] to implement SVM. Using machine learning methods allows us to implement multiple detection methods by choosing different sets of features as training data. The features used in the experiment were:

Rule-based features: **(a)** identical match of phonemes in seed and transformed expressions, **(b)** handling alternation of phonemes with phonological similarity, **(c)** allowance of deletion/addition of repetitive consonant phoneme added to (b).

Machine learning-oriented features: **(d)** Bag-of-Words features.

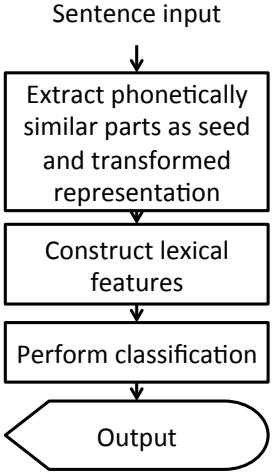

Figure 2: The outline of the proposed pun detection method.

3.1 Rule-based Features

Feature (a): A reading *Kana* sequence is extracted using a morphological analyzer *MeCab*[2] and *JUMAN*[3]. Here, each token or sequence of them is treated as a **seed** expression. The feature value is set to 1.0 when the pronunciation of the reading *Kana* is identically reproduced as transformed expression. However, in order to prevent erroneous detection due to a large number of matching sections for a reading *Kana* of one character, the target reading *Kana* shall be two or more characters.

Feature (b) and (c): In addition, since imperfect juxtaposed pun with a part of the vocal sounds transformed occupies more than 85.0% of targeted puns, we consider that absorption of different pronunciations is necessary for detecting phonemic similar parts. Therefore, we constructed features **(b)** and **(c)** using phonological similarity of consonants described later. Specifically, the following processing is added to the reading *Kana* sequence extracted in the feature (a). First, in order to avoid the consistency of single words, we divide the reading *Kana* into mora units and arrange it. At this time the method converts the long tone to diphthong moras and give the mora of geminate and repellent unique signs. The consonant phonemes and vowel phonemes in the mora are separated and stored in one array.

[1] http://scikit-learn.org
[2] http://taku910.github.io/mecab/
[3] http: //nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

For example, *intaashoppu* (intershop) is converted into an array such as `{/*,i/, /*,N/, /t,a/, /*,a/, /sh,o/, /*,Q/, /p,u/}`. Here, `N` represents sound absorption, and `Q` represents geminate. The notation of phonemes of other consonant parts is the same as the phonetic expression by Hepburn type *Romaji*.

The system calculates consonant similarity for different consonants p and q using the function

$$s_c(p,q) = \frac{N_{\text{pair}}\, N_{p,q}}{N_p\, N_q}. \quad (1)$$

Here N_p and N_q are the concurrence frequency of the two consonants in a seed and transformed expressions respectively, or vice versa, and N_{pair} the number of all the consonant pairs.

For example, when calculating phonological similarity of a pair of expression `{/n,i/, /*,N/, /gy,o/}` and `{/n,i/, /*,N/, /j,o/}`, the consonant pair, `/j/` is used. If using the development data constructed in Section 4, $s_c(/\text{gy}/,/\text{j}/) = 0.1$. And if there are multiple transformed expression candidates for one type expression candidate, we treat all transformed expression candidates as equivalents rather than the one with a higher value.

In addition, even when the pluralities of transformed expression candidates become highest, all the transformed expression candidates are detected accordingly. In the condition of detection of seed / transformed expression, the number of detections within one sentence of the seed / transformed expression and the magnitude of the similarity value is not taken into consideration. However, we consider that these indicators give superficial information indispensable for automatic evaluation of recognition of the meaning of a pun and how fun it is.

3.2 Machine Learning-oriented Features

Features (d): The proposed method creates Bag-of-Words features by extracting the stem of a content word (noun, proper noun, general verb, adjective, adverb, emotive verb) using MeCab as the morphological analyzer, and IPAdic (Asahara and Matsumoto, 2003) the morpheme dictionary.

The values of Bag-of-Words features are determined as follows. We associate Bag-of-Words acquired from all input sentences of training data in advance with feature numbers. At the time of learning and detection, similarly the method obtains a Bag-of-Words from input sentences. Finally, the method lets the value of the feature corresponding to the content word to be 1.0.

4. Detection Experiment and Results

For the experiment, negative example data consisting of sentences not including puns was prepared. We extracted the sentences for the data from the YACIS corpus (Ptaszynski et al., 2012), which collected blog articles in Japanese posted between 2011 and 2012. We randomly extracted the sentences

In actual data such as chat dialogue corpus, the number of positive examples is considered to be extremely small compared to the number of negative examples. In such a case, it is necessary to be able to suppress deterioration of detection performance against robustness reduction (robustness). In order to compare the robustness of each feature set, we used the feature sets (shown in Table 3) F8, F9, F10, F11, F12, F13 and F14. These feature sets showed particularly high detection performance in preliminary experiments. We performed a 20-fold cross validation for evaluation data, changing the ratio of positive data number to negative one. The ratio was repeatedly increased by 0.01, from 0.01 to 1.0. Figure 3 shows the plot of the positive example negative example ratio in this result on the horizontal axis and the F-measure showing the detection performance on the vertical axis.

Due to constraints of computing resources, we used linear kernel as a kernel function for this observation.

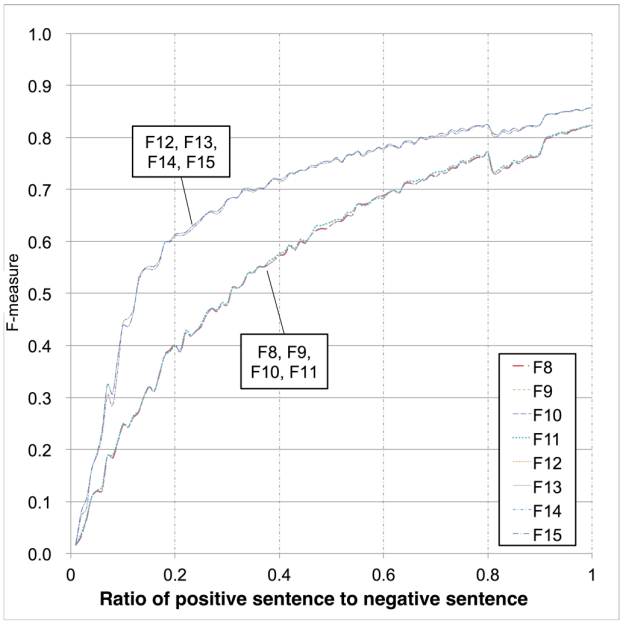

Figure 3: A plot of F-measure scores to varied sizes of positive sentences for each feature set.

FS	(a)	(b)	(c)	(d)	FS	(a)	(b)	(c)	(d)
F1	-	-	✓	-	F9	-		✓	✓
F2	-	✓	-	-	F10	-	✓	-	✓
F3	-	✓	✓	-	F11	-	✓	✓	✓
F4	✓	-	-	-	F12	✓	-	-	✓
F5	✓	-	✓	-	F13	✓	-	✓	✓
F6	✓	✓	-	-	F14	✓	✓	-	✓
F7	✓	✓	✓	-	F15	✓	✓	✓	✓
F8	-	-	-	✓					

Table 3: Feature sets and features for each set.

Kernel	FS	F-measure	Accuracy
Linear	F15	0.858	0.857
RBF	F15	0.908	0.909

Table 4: The feature set that showed highest F-measure score for RBF and Linear kernels.

Using the same data as the sample for categorization shown in Table 1 and Section 1, we tuned parameters of the SVM and its kernels.

5. Discussion

5.1 Effectiveness of SVM with RBF Kernel

In the experiment comparing performance between RBF and Linear kernels, the results in Table 4 show that the RBF kernel has higher performance. However, the RBF kernel required a lot of time for learning. In the experiment we have conducted, using a Xeon 2.0GHz CPU with around 64GB free memory, one learning division ended in 3,753 and 2,420 seconds, respectively. On the other hand, the learning using the linear kernel took only 6.54 seconds.

Hence, it can be said that the performance with respect to the unit requirement of the computational resource becomes maximum in the case of the linear kernel. However, there still remains a strong need for a faster learning algorithm as accurate as an SVM with RBF Kernel.

5.2 Adding Rule-based Features

As shown in Figure 3, in the experiment using the linear kernel, there was no significant difference between the feature sets F12, F13, F14, F15 and F8, F9, F10, F11. The difference between F8 vs. F10, F9 vs. F11, F12 vs. F14 and F13 vs. F15 is the addition of phonological similarity feature (b). While there was a factor of the significant difference, but no significant difference was observed. The same applies to the other phonological similarity feature (c) that allows insertion of acoustic long sounds. (Compare F8 to F9, F10 to F11, F12 to F13, and F14 to F15.) In the feature (c), a significant difference was confirmed in two sets of F4 to F5 and F6 to F7, though this was not tested in the experiment in this experiment's scale. It is understood that the effects of the features (b) and (c) were not remarkable in this experiment.

On the contrary, the features showing significant differences by addition of features are (a) perfect match features (F8 to F12, F9 to F13, F10 to F14, and F11 to F15). From this fact, although there is room for improvement in the features (b) and (c), compared with the case of only lexical features (d) (feature set F8) due to addition of rule base feature (a). It can be said that the robustness against reduction of the number of positive data is improved.

6. Conclusion and Future Work

In this research, we proposed a method to detection of similar parts using phonological similarity and insertion / omission of prolonged sounds in addition to lexical feature in detection of sentences including pun, based on supervised learning by SVM. In the detection performance evaluation experiments, we confirmed the effectiveness of lexical feature, the validity of rule base feature and each case of successful detection, respectively, and showed the overall effectiveness of the proposed method alongside the phonological nature of sentences. Moreover, by using lexical feature, it was shown that the proposed method obtains constant detection performance against overlapping pun in addition to collocation type puns. However, in considering the results, there is room for improvement in rule base identification, and the dependence of lexical feature on learning data was suggested. Also, we have found that the scale around 50,000 sentences of Japanese puns is suitable for construction of its detection method.

As one of current work, we are doing precise research of more optimized detection method including the use of neural language models. We are going to do investigations of the dataset itself in order to comprehend the linguistic nature of puns quantitatively.

7. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP17K00294.

8. Bibliographical References

Yamada, K., Asano, S., and Mononobe, H. (2012). A Study on the Relationships among Self-Efficacy, Social Skills, Assertive Communication Skill, and Coping Humor in Interpersonal Conflict. *The Japanese Journal of School Health*, 54:203--210. (In Japanese)

Miyazawa, K., Tokoyo T., Masui Y., Matsuo, N., and Kikuchi, H. (2012). Factors of Interaction in the Spoken Dialogue System with High Desire of Sustainability. *The journal of the Institute of Electronics, Information and Communication Engineers*. J95-A(1): 27-36. (In Japanese)

Takizawa, O., (1995), Several Phonemic Features of Written Puns. *Journal of Natural Language Processing* (ANLP), 2(2):3-22. (In Japanese)

Kawahara, S. and Shinohara K., (2009). The role of psychoacoustic similarity in Japanese puns: A corpus study. *Journal of Linguistics*, 45(1):111-138.

Tsukawaki, R., Fukada, H., Higuchi, M., (2011). Process effects of expression of humor on anxiety and depression. *The Japanese Journal of Experimental Social Psychology*, 51(1):43--51. (In Japanese)

Dybala, P., Ptaszynski, M., Jacek, M., Takahashi, M., Rzepka, R., and Araki, K., (2012). Multiagent system for joke generation: Humor and emotions combined in human-agent. *Journal of Ambient Intelligence and Smart Environments*, 2(1):31-38.

9. Language Resource References

Ptaszynski, M., Dybala P., Rzepka R., Araki, K., and Momouchi, Y. (2012). A Five-Billion-Word Corpus of Japanese Blogs Fully Annotated with Syntactic and Affective Information. In Proc. of LaCATODA, a workshop of the AISB/IACAP World Congress in Honour of Alan Turing, pp. 40-49, Birmingham, UK.

Araki, K., Uchida, Y., Sayama, K., and Yatsu., M. (2017). Construction and Analysis of Pun Database in Japanese. In Proc. of the 56th SIG. Language Sense processing and Engineering (SIG-LSE-B702-3), pp. 13-24, Iwate, Japan. (In Japanese)

Asahara, M., and Matsumoto, Y., (2003). IPADIC User Manual. Nara Institute of Science and Technology, Japan.