Jujeop: Korean Puns for K-pop Stars on Social Media

Soyoung Oh*
Sungkyunkwan University, Korea
sori424@g.skku.edu

Jisu Kim*
Sungkyunkwan University, Korea
rlawltn908@g.skku.edu

Seungpeel Lee*
Sungkyunkwan University, Korea
Sahoipyoungnon Publishing, Korea
leepeel@g.skku.edu

Eunil Park†
Sungkyunkwan University, Korea
and Raon Data, Korea
eunilpark@skku.edu

Abstract

Jujeop is a way for K-pop fans to express their love for the K-pop stars they adore by creating a type of Korean pun through unique comments in *Youtube* videos that feature those K-pop stars. One of the unique characteristics of *Jujeop* is its use of exaggerated expressions to compliment K-pop stars, which contain or lead to humor. Based on this characteristic, *Jujeop* can be separated into four distinct types, with their own lexical collocations: (1) Fragmenting words to create a twist, (2) Homophones and homographs, (3) Repetition, and (4) Nonsense. Thus, the current study defines the concept of *Jujeop* and manually annotates the 8.6K comments into one of the four *Jujeop* types. With the given annotated corpus, this study presents distinctive characteristics of *Jujeop* comments compared to the other comments by classification task. Moreover, with the clustering approach, we proposed a structural dependency within each *Jujeop* type. We have made our dataset publicly available for future research of *Jujeop* expressions.

1 Introduction

With the rapid improvement of information and telecommunication technologies, people have become not only consumers, but also producers of media content (Jenkins and Deuze, 2008). With this trend, there are a number of online media platforms that allow people to interact with other users anywhere and anytime (Burgess and Green, 2018). On these platforms, users actively create and share their contents, and express their thoughts and opinions on other users' contents (Van Dijck, 2013). In particular, online fan communities, where fans interact with each other, tend to use such platforms to share their contents and opinions on their favorite stars (e.g., Ariana Grande[1], BTS[2]; Baym (2007); Littlejohn and Foss (2009)).

With this vitalization of the communities on the platforms, several novel interaction patterns have been observed among South Korean users. Among these patterns, *Jujeop* in online environments is one of the notable phenomena presented by South Korean fans (Figure 1). Although the dictionary definition of the Korean word *Jujeop* refers to a disgraceful or silly behavior of a person, the term has evolved into a facetious expression with an implicit sense of humor in the online K-pop community; in South Korean culture, *Jujeop* is a punning activity that makes conversations enjoyable and allows users to engage on platforms (Yu et al., 2018).

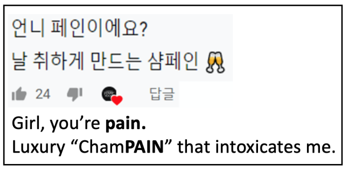


Figure 1: An example of *Jujeop* comments on *Youtube*

Miller and his colleagues (Miller et al., 2017) defined a pun as "*a form of wordplay in which a word suggests two or more meanings by exploiting polysemy, homonymy, or phonological similarity to another word, for an intended humorous or rhetorical effect.*" Based on this definition, the majority of recent studies have proposed several pun generation models using machine learning approaches (He et al., 2019; Luo et al., 2019).

However, compared to a huge body of prior research on English puns (Yu et al., 2018, 2020), only a few studies have been conducted on Korean puns in online environments. Because of some obstacles including the unique linguistic and cultural aspects of South Korea, there are several limitations in studying users' punning activities (Choi, 2018).

Thus, we propose the first Korean corpus, annotated for *Jujeop* comments, and categorize them into four different types. We have made the dataset

* Equally contributed first authors
† Corresponding author
[1] https://rb.gy/mzl1vq
[2] https://rb.gy/0dfcdl

Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media, pages 170–177
Online Workshop, June 10, 2021.

publicly available.[3]

2 *Jujeop* Data

2.1 Data Collection

As *Jujeop* comments are frequently observed in *Youtube* channels of K-pop stars, we assumed that high number of views in a channel guarantees the presence of the *Jujeop* comments. Based on this assumption, we collected 281,968 users' comments on K-pop stars from 285 *Youtube* channels[4], which have the number of views between 5,177 and 38,039,597. Then, we conducted the pre-processing procedures for the remaining Korean words (i.e., excluding words used for commercial purposes).

We sorted the comments based on the number of likes a *Jujeop* comment received. The comments that had more than the average number of likes in the collected comments (i.e., 167) were employed. With this approach, 8,650 comments were selected for annotation.

2.2 Annotation

Ten annotators who has been enthusiastic fans of their K-pop stars for at least 2 to 15 years (Mean: 9.3 SD: 4.2) and has been frequently exposed to *Jujeop* comments were employed for the annotation process. After explaining the definition and examples of *Jujeop* comments, each annotator was asked to respond to the following question to classify, whether each comment is a *Jujeop* comment:

- Is this a *Jujeop* comment, which has a sense of humor by praising K-pop stars with exaggerations and flashy modifiers?

Then, each annotator was asked to classify the *Jujeop* comment into one of the following types.

2.2.1 Fragmenting words to create a twist

The comments in this type intentionally fragment a specific word and extract/concentrate a single character from the word to disguise the word's full meaning (e.g., 'pretty' to 't'), in order to create a twist in the sentence meaning.

When one of the characters is included in both a specific word and sentence with the same pronunciation, the word and sentence are linked. This means that there are two steps in a *Jujeop* comment. After the sentence with hidden or sarcastic meanings is first presented, the word with complimentary meanings is then provided. For instance, 't' can mean 'tee' (t-shirt) as it has the same Korean pronunciation. Moreover, the fragmented word (e.g., 'T') usually carries a neutral connotation, while the complete word (e.g., 'Pretty') carries a positive connotation.

Because two words are linked and combined to make a sentence ('t' (t-shirt) and 'pretty'), it creates a pun in Korean:

언니. 왜 맨날 똑같은 **티**만 입어? **프리티!**
Sis, Why do you always wear the same **Tee**? **pretTee!**

The first sentence asks why she always wears the same t-shirt, which is pronounced [ti:]. Then, the following word changes the whole sentence meaning, which makes the initial meaning of the sentence a compliment about her prettiness [prti], thus creating a humorous twist.

2.2.2 Homophones and Homographs

Both homophones and homographs are sometimes employed to create pun expressions.

Homophones are defined as follows: "*when two or more words, different in origin and signification, are pronounced alike, whether they are alike or not in their spelling, they are said to be homophones*" (Bridges, 2018). The definition of homographs is "*words that have more than one meaning but share the same orthography*" (Twilley et al., 1994).

Users can employ specific lexical features of homophones and homographs to make a *Jujeop* comment. After a user makes his/her first sentence with the original meanings of words, they employ other word meanings in the second sentence to compliment the K-pop stars while allowing other users to enjoy the fun.

For example, George Bush, the former US president, has the same pronunciation in Korean and English (Korean: '조지 부시'), when George Bush is employed as a big name. The South Korean pronunciations of George is identical to the phrase 'to beat somebody/something' (Korean: '조지(다)'), while the pronunciation of Bush is identical to 'to break something' (Korean: '부시(다)'). Thus, the pronunciations of George Bush and 'to beat somebody/something + to break something' can be the same in Korean, although the meanings of the words differ depending on whether they are employed as a big name or as verbs.

너 영어이름을 **조지 부시**로 해줘...
내 마음을 **조지고 부시**니까.
Change your English name to **George Bush**...
because you **beat and break** my heart.

[3] https://github.com/merry555/Jujeop

[4] https://github.com/merry555/Jujeop/blob/main/dataset/channels.txt

2.2.3 Repetition

This is a type of repetition of the same phrase. As presented in the following example, the comments in this type employ repetition to emphasize the complimentary meanings on the K-pop stars.

아 진짜.. 그거 알아요? 잘생긴 사람을 보면 기억을 잃는대요.
아 진짜.. 그거 알아요? 잘생긴 사람을 보면 기억을 잃는대요.
Gosh... you know what? They say you lose your memory when you see a handsome person.
Gosh... you know what? They say you lose your memory when you see a handsome person.

2.2.4 Nonsense

The comments in this type include the K-pop stars within fictions. The majority of such comments flatter the stars by using exaggerated and almost nonsensical, over-the-top expressions. One representative example is presented below:

그녀가 예쁘다고 생각하는 사람 일어나! 라고 했더니 **지구가 일어나서 태양계 순서가 바뀌었잖아.**
I said, **Anybody who thinks she's pretty, get up!** and then **the whole Earth got up and the order of the solar system changed.**

There is no way that the Earth can 'get up' like a human being, nor could the order of the solar system change due to a person's prettiness. Such ridiculous and exaggerated expressions create humor and a profound expression with which fans can express admiration for their favorite celebrities.

2.3 Corpus Description

Among 8,650 comments, 1,867 (21.58%) were annotated as *Jujeop* comments. Then, three experts in natural language processing (NLP) manually validated whether or not each comment is a *Jujeop* comment. With these procedures, 7,077 non-*Jujeop* (81.82%), and 1,573 *Jujeop* (18.18%) comments were labeled with four separate *Jujeop* types (Table 1). We measured Krippendorff's alpha on four types of *Jujeop* comments (Krippendorff, 2011), and met inter-annotator agreement (0.532).

Type	Count
Fragmenting words to create a twist	39
Homophones and Homographs	57
Repetition	41
Nonsense	1436

Table 1: Descriptive analysis of *Jujeop* comments.

3 Experiments

We conducted two NLP tasks to investigate whether the labeled data can be significant in understanding *Jujeop* comments. First, we proposed several deep learning models to verify the annotated *Jujeop* comments. Then, we clustered *Jujeop* comments to figure out specific linguistic structures.

3.1 *Jujeop* Classification

At first, for the *Jujeop* classification, we applied three baseline classifiers for the experiment: Convolutional Neural Network (CNN; Kalchbrenner et al. (2014)), Bidirectional Long Short-Term Memory (BiLSTM; Schuster and Paliwal (1997)), and KoBERT[5]. All model configurations are presented in Appendix A.

Because more than 80% of the annotated comments in the dataset are non-*Jujeop* comments, we randomly selected 1,573 non-*Jujeop* comments, which is the same number of *Jujeop* comments to address the data imbalance issue. Then, we randomly divided the collected comments into training (2,256, 72%), validation (260, 8%), and testing (630, 20%) sets. We tokenized each comment with the *Mecab* tokenizer of *KoNLPy* package[6]. The maximum word counts of the comments and total vocabulary size are 58 and 6,536, respectively.

Classifier	Class	Precision	Recall	F1-score	Accuracy
CNN	*Jujeop*	75.41%	72.44%	73.90%	69.05%
	non-*Jujeop*	60.23%	63.86%	61.99%	
BiLSTM	*Jujeop*	77.59%	72.70%	75.07%	70.79%
	non-*Jujeop*	61.90%	67.87%	64.75%	
KoBERT	*Jujeop*	**80.45%**	**74.54%**	**77.38%**	**73.65%**
	non-*Jujeop*	**64.98%**	**72.29%**	**68.44%**	

Table 2: Results of the binary classification task (*Jujeop* and non-*Jujeop* comments).

F1-score	*Jujeop* (*2-ary*)	*Jujeop* type (*4-ary*)
CNN	67.94%	62.63%
BiLSTM	69.91%	56.96%
KoBERT	**72.91%**	**77.18%**

Table 3: Results of the macro f1-score; *2-ary*: binary classification of *Jujeop* and non-*Jujeop*, *4-ary*: multi-class classification of *Jujeop* types.

Table 2 presents the classification results with four evaluation metrics. In general, the KoBERT showed the greatest levels of all evaluation metrics. In particular, the accuracy of the KoBERT (73.65%) was higher than those of the CNN (69.05%) and BiLSTM (70.79%). In case of the recall level of

[5] https://github.com/SKTBrain/KoBERT
[6] https://konlpy.org/ko/v0.4.3/api/konlpy.tag/

Jujeop comments, it can be explained by the potentiality of misclassifying *Jujeop* to non-*Jujeop* comments. Moreover, we measured macro F1-score for the binary classification task (Table 3). Compared to the other benchmark models, KoBERT showed the best performance (72.91%).

Furthermore, we computed macro F1-score for the *Jujeop* classification task as each type of comment had a skewed distribution (Tran et al., 2018). The details of configurations are attached on Appendix A. Table 3 shows KoBERT with the highest performance of 77.18% followed by CNN (62.63%) and BiLSTM (56.96%). The implemented models are publicly available[7].

3.2 *Jujeop* Clustering

Pun usually relies on specific linguistic structure that can be classified based to patterns of the syllable, word, or phrase similarity (Binsted and Ritchie, 1997; Ritchie et al., 2007). Since, *Jujeop* comments share the characteristic of the pun, we assumed that *Jujeop* comments within the same type would share similar dependency relations.

Based on the assumption, we employed part-of-speech (pos) tagging to analyze the distinctive linguistic structure of each *Jujeop* type. Then, the tagged sentences were used as the input for the unsupervised learning algorithm, which allows identification of data into similar groups or clusters (Likas et al., 2003).

We utilized *Okt* pos tagger, which is commonly used to analyze the social media data analyses (Park and Cho, 2014). First, to balance the number of each type in *Jujeop* comments, we randomly selected 50 samples from type 4. Then, we vectorized each pos tag of the sentence as an input to the K-means clustering with K as 4, which represents 4 types of *Jujeop* comments.

Figure 2 represents the confusion matrix of the true and the predicted data points. The total accuracy of the K-means clustering was 32%, where the most correctly predicted type was type 2 with the 34 out of 57 correct predictions (59.65%).

Whereas most of type 1 were classified into type 3 (23 out of 39), which indicates that two types might share similar dependency relations. The single word appeared at the beginning of the sentence that was used again at the later part might have been characterized as a repetition. Type 3 was classified with 48.78% accuracy (20 out of 51), which indicates that type 3 might have been differentiated by syntactic features with the other types.

[7]https://github.com/merry555/Jujeop/tree/main/models/multiclass

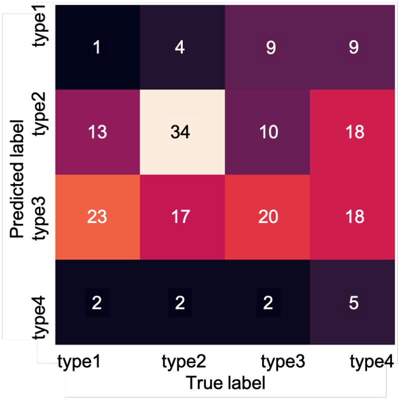


Figure 2: Confusion matrix on the clustering results of *Jujeop* types; x-axis indicates the true *Jujeop* types and y-axis indicates the predicted *Jujeop* types

Moreover, type 4 showed the lowest clustering accuracy with 10% (5 out of 50). This indicates that nonsense might be interpreted as semantic feature rather than syntactic feature. The further explanations and visual supplements are attached in Appendix B.

4 Conclusion

The current study first conceptualized the construct of *Jujeop*, which is one of the Korean pun interaction patterns on social media and annotated 8,650 comments. To provide a better understanding of *Jujeop* comments, four separate *Jujeop* types were proposed and labeled. Then, the presented NLP tasks results imply that *Jujeop* comments and each type of *Jujeop* has semantic and syntactic distinctiveness compared to the other comments.

Although we provide several findings on *Jujeop* comments, notable limitations remain. First, there are limited number of each type of *Jujeop* comments. Moreover, there might be other *Jujeop* types that were not observed in this study. The presented limitations might have occurred from the fact that the examples of *Jujeop* may be hard to collect in the wild. Thus, future study should aim to overcome the presented limitations with a crowd sourcing experiment or sentence generation based on the given definition to make a corpora of various *Jujeop* comments.

Acknowledgements

Special thanks to Eun Been Choi and Jihwan Aum for their helpful comments. We would also like to thank “사회평론” (Sahoipyoungnon Publishing) for providing financial support with annotation. This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program (IITP-2020-0-01816) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation). This research was also supported by the National Research Foundation of Korea funded by the Korean Government (NRF-2020R1C1C1004324).

References

Nancy K Baym. 2007. The new shape of online community: The example of swedish independent music fandom. *First Monday*, 12(8).

Kim Binsted and Graeme Ritchie. 1997. Computational rules for generating punning riddles. *Humor: International Journal of Humor Research*, 10(1):25–76.

Robert Bridges. 2018. *On English Homophones*. Litres.

Jean Burgess and Joshua Green. 2018. *YouTube: Online video and participatory culture*. John Wiley & Sons, Cambridge, United Kingdom.

Jinsook Choi. 2018. A linguistic anthropological study of the typification of middle-aged men in korea: An examination of ajae joke data. *Korean cultural anthropology*, 2:109–139.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1734–1744.

Henry Jenkins and Mark Deuze. 2008. *Convergence culture*. Sage Publications, London, United Kingdom.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665.

Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability. https://repository.upenn.edu/asc_papers/43/.

Friedrich Leisch. 2006. A toolbox for k-centroids cluster analysis. *Computational statistics & data analysis*, 51(2):526–544.

Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461.

Stephen W Littlejohn and Karen A Foss. 2009. *Encyclopedia of communication theory*, volume 1. Sage Publications, London, United Kingdom.

Fuli Luo, Shunyao Li, Pengcheng Yang, Lei Li, Baobao Chang, Zhifang Sui, and SUN Xu. 2019. Pun-gan: Generative adversarial network for pun generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3379–3384.

Tristan Miller, Christian F Hempelmann, and Iryna Gurevych. 2017. Semeval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68.

Eunjeong L. Park and Sungzoon Cho. 2014. Konlpy: Korean natural language processing in python. In *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*, Chuncheon, Korea.

Graeme Ritchie, Ruli Manurung, Helen Pain, Annalu Waller, Rolf Black, and Dave O’Mara. 2007. A practical application of computational humour. In *Proceedings of the 4th International Joint Conference on Computational Creativity*, pages 91–98.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Duc Tran, Hieu Mac, Van Tong, Hai Anh Tran, and Linh Giang Nguyen. 2018. A lstm based framework for handling multiclass imbalance in dga botnet detection. *Neurocomputing*, 275:2401–2413.

Leslie C Twilley, Peter Dixon, Dean Taylor, and Karen Clark. 1994. University of alberta norms of relative meaning frequency for 566 homographs. *Memory & Cognition*, 22(1):111–126.

José Van Dijck. 2013. *The culture of connectivity: A critical history of social media*. Oxford University Press, Oxford, United Kingdom.

Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660.

Zhiwei Yu, Hongyu Zang, and Xiaojun Wan. 2020. Homophonic pun generation with lexically constrained rewriting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2870–2876.

A Model Configuration

A.1 CNN

A.1.1 Binary classification

To employ a CNN-based classifier, we created a sequence of the tokenized words by embedding a layer with 128 units. The sequence was then sent to the CNN layer with 64 units. The max pooling layer was used to extract the prominent features of the given data. The final output was computed by sigmoid function to classify whether or not the given comment is a *Jujeop* comment. Ten epochs were employed in the training sessions with 32 batch size.

A.1.2 Quaternary classification

We used the the same configurations with the binary classification task except optimizer, loss and activation functions of the last layer. For the multi-class classifiction task, we employed the softmax activation function for the last layer and sparse categorical crossentropy for the loss function with adam optimizer. Also, we compiled the model with class weights by scikit-learn package[8] to handle the class imbalance problem.

A.2 BiLSTM

A.2.1 Binary classification

The tokenized words of the comments were outputted to the embedding layer with 128 units. The representation of the input data was then sent to the bi-directional LSTM layer with 64 units. The final output of the BiLSTM was calculated through sigmoid function. We trained the model with 10 epochs with 256 batch size.

A.2.2 Quaternary classification

We changed the optimizer, loss and activation functions of the last layer as in a CNN classifier for the multi-class classification. We also compiled the model with same class weights as in the CNN classifier.

A.3 KoBERT

A.3.1 Binary classification

To employ a KoBERT model, we adopted a built in SentencePiece tokenizer. We set embedding size as 128 and trained the model with 10 epochs. We set the batch size as 32 and learning rate as 0.00002.

A.3.2 Quaternary classification

We used same configurations as in the binary classification task. For the multi-class classification task, we modified the class number of the KoBERT classifier to 4.

B Jujeop Clustering

B.1 K-means Clusters Visualization

As shown in Figure 3, we visualized each type of *Jujeop* clusters with predicted data types and true data types. The predicted clusters are the results from K-means clustering with pos tagged *Jujeop* comments.

B.2 Centroids of the clusters from all types

Based on the K-means clustering results, we analyzed the dependency trees of centroids which are the representative data points to separate each cluster (Leisch, 2006). The structure of the type 1 centroid presents as below:

언니 다 좋은데 자꾸 **벽**이 느껴져요 **완벽**.
Sis, you make a **wall**. A **Perfection**.

[(NP<언니, Noun>) (AP <다, Adverb> <좋은데, Adjective>) (NP <자꾸, Noun> <벽,Noun>) 이,Josa (VP<느껴져요,Verb>) (NP<완벽,Noun>)]

which fragments word "벽" to make the word "완벽" to convert the meaning of the word "wall" into "perfection".

Moreover, the center data point of the type 2 is proposed as below:

언니 경**마**장 가지마요 언니가 경마장가면 **말**이 안나와.
Sis, don't go to **horse-racing**. Because you are **horse-less**.

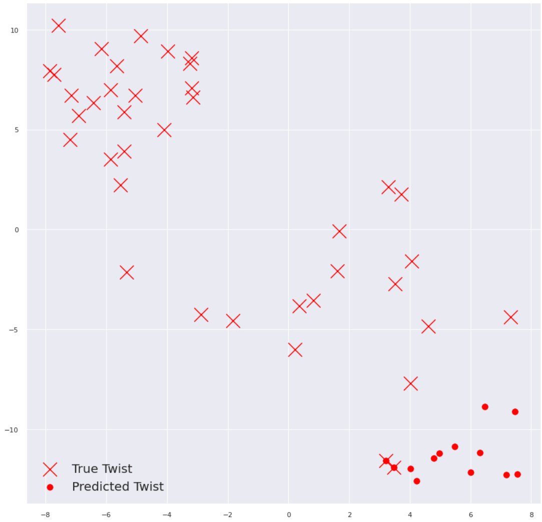
[(NP<언니,Noun> <경마장,Noun> <가지,Noun> <마,Noun>) 요,Josa (NP<언니,Noun>) 가,Josa (NP<경마장,Noun> <가면,Noun> <말,Noun>) 이,Josa (NP<안나,Noun>) 와,Josa]

where the English word "horse" has the same pronunciation as "speech" in Korean as "말". Based on this homophone effect of word "말" in Korean, the horse-less can be interpreted as speechless.

The centroid of the type 3 is represented as below:

듣다 눈물날것같음 전남친이 저렇게 날 예쁘게 회상해준다면... 난 사실 **전남친 없어요, 남친도 없어요, 없어요, 아니 없어요, 그냥 없어요.**

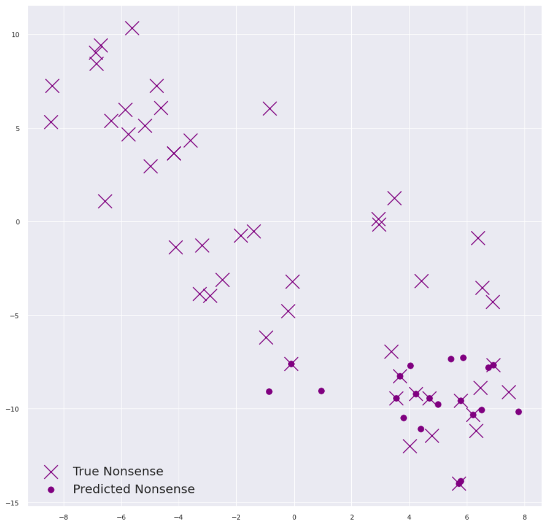
[8] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html



(a) Clusters of Type 1: Fragmenting words to create a twist

(b) Clusters of Type 2: Homophones and Homographs

(c) Clusters of Type 3: Repetition



(d) Clusters of Type 4: Nonsense

Figure 3: K-means clustering results of each type of *Jujeop* comments where K as 4; Marker ○ as predicted data points and Marker × as true data points

I'm going to cry if my ex-boyfriend recalls me so beautifully... Actually I have **no ex, no boyfriend, nothing, nothing, nothing, just nothing**.

[(VP <듣다,Verb> <눈물날것,Verb>) (AP <같음,Adjective>) (NP <전남친,Noun>) 이,Josa (AP <저렇게,Adverb>) (NP <날,Noun>)(AP <예쁘게,Adjective>)(NP <회상,Noun>) (VP <해준다면,Verb>) (NP <난,Noun> <사실,Noun> <전남친,Noun>) (AP <없어요,Adjective>) (NP <남친,Noun>) 도,Josa (AP <없어요,Adjective> <없어요,Adjective> <아니,Adjective> <없어요,Adjective>) (NP <그냥,Noun>) (AP <없어요,Adjective>)]

which repeats the same word of "nothing" to make humor with emphasizing the attraction of the K-pop stars, simultaneously.

Moreover, the most representative data point of type 4 is given as below:

어이없네 이런걸 노래라구 낸건가. 그냥 이나은 인생 주제곡이잖아. 요즘 아이돌들 참 쉽다. 성의 없네. 그냥 이 노래 자체가 이나은인디.
It is ridiculous that this can be called as a song. It's just a life of "Naeun Lee". How easy to become star these days. This song is as "Naeun Lee" itself.

[(AP<어이없네,Adjective>) 이런,Modifier (NP <걸,Noun> <노래,Noun>) 라,Josa (NP <구,Noun>) (VP <낸,Verb>) (NP <건가,Noun> <그냥,Noun>) 이,Determiner (NP <나은,Noun> <인생,Noun> <주제곡,Noun>) 이,Josa (VP <잖아,Verb>) (NP <요즘,Noun> <아이돌,Noun> <들,Suffix>) (VP <참,Verb> <쉽다,Verb>) (NP <성의,Noun>) (AP<없네,Adjective>) (NP <그냥,Noun> <이,Noun> <

노래,Noun> <자체,Noun>) 가,Josa 이,Determiner (NP <나은,Noun>) 인,Josa (NP <디,Noun>)]

which is far from the defined nonsense comments as it doesn't contain any of the nonsensical features. Rather, the presented centeroid comment uses critical note to paradoxically emphasize the coolness of the k-pop star. Considering the fallacious unsupervised classification results of type 4 (Figure 2), the given type would be interpret with semantic meanings rather than syntactic relations.