How many kaiju are needed to (dialectically) fight King Kong? Development of a first-two-characters input method

Poveda Gutiérrez Santiago
Langiage Information Processing, Advanced
santiago.gutierrez.23m@st.kyoto-u.ac.jp

Code and data can be found in the Afitwchinkame GitHub repository.

Abstract

The realm of kaiju content on platforms like YouTube is rapidly expanding, driven by the enthusiasm of fans who create intricate narratives and analyses of their favorite monsters. However, the recent influx of low-quality AI-generated content has threatened the originality and depth that are hallmarks of this niche. To address this issue, we have developed an autocompletion model tailored specifically for kaiju content creators. This tool aims to elevate the quality of user-generated content by providing contextually accurate and engaging script completions, thereby supporting the community of kaiju enthusiasts in their creative endeavors. By leveraging Support Vector Machines (SVMs) and focusing on linguistic battles rather than physical confrontations, we explore the rhetorical capabilities of kaiju. Our findings demonstrate that a training set comprising texts from just four kaiju suffices to match King Kong's eloquence in a dialectic duel. Our study highlights the potential of SVM-based autocompletion models to enhance niche content creation and opens new avenues for research in kaiju linguistics.

Key words: Kaiju, Corpus Creation, KyTea, SVM, L1-regularization, L2-regularization, Performance Measures, Edit distance, NLP

1 INTRODUCTION

The realm of kaiju content on platforms like YouTube is rapidly expanding, driven by the enthusiasm of fans who create intricate narratives and analyses of their favorite monsters. However, the recent influx of low-quality AI-generated content has threatened the originality and depth that are hallmarks of this niche. To address this issue, we have developed an autocompletion model tailored specifically for kaiju content creators. This tool aims to elevate the quality of user-generated content by providing contextually accurate and engaging script completions, thereby supporting the community of kaiju enthusiasts in their creative endeavors.

While there is an abundance of works that pit Japanese kaiju against King Kong in epic battles of strength ((Honda, 1962), (Honda, 1967), (Wingard, 2021), (Wingard, 2024)), the academic exploration of dialectic encounters between these giants remains unexplored. Our work seeks to bridge this gap by investigating the linguistic prowess of various kaiju through autocompletion models. Specifically, we aim to develop autocompletion models that take only the first two characters of each word (token) as input, and then output a text by completing each word. By focusing on dialectic rather than physical confrontations, we open a new dimension of kaiju research that emphasizes wit and rhetoric over brute force.

A critical aspect of our research is determining the number of kaiju required to match the eloquence of King Kong in a dialectic duel. In other words, how much linguistic information from classic Japanese kaiju is necessary to accurately predict the content of the King Kong Wikipedia article, which we used as testing data. This measurement is essential for understanding the narrative potential of kaiju and the ability of autocompletion models to create meaningful discourses. Our models, trained on a diverse range of kaiju-related texts, provide insights into the linguistic capabilities of these monsters, offering a fresh perspective on their roles in creative content.

On another hand, Support Vector Machines

(SVMs) are particularly well-suited for our study due to their effectiveness with relatively small datasets (Awad and Khanna, 2015). Unlike other machine learning models that require vast amounts of data, SVMs can deliver high performance even when trained on limited corpora, making them ideal for niche applications like ours. By employing SVMs, we demonstrate the feasibility of creating robust autocompletion tools for specialized content creation, paving the way for future research in this domain.

This paper is structured as follows: We first outline the methodology used to create and train our SVM-based autocompletion models. After that, we explain the process of creating our own kaiju corpora and the specifications for the databases used for training and evaluation. Next, we present the results of our experiments, highlighting the performance of each model on the KINGKONG dataset. Finally, we discuss the implications of our findings for the kaiju content community and suggest directions for future research.

2 METHOD

In this study, we have trained and compared L1-regularized and L2-regularized trainable SVM models using the LIBLINEAR library (Fan et al., 2008). To facilitate the training and testing processes, we utilized the Kyoto Text Analysis Toolkit (KyTea) (Neubig et al., 2011), a powerful tool designed for text processing and analysis.

Support Vector Machines (SVMs) are supervised learning models used for classification and regression tasks. They work by finding the hyperplane that best separates the data into different classes. Given a set of training data, each labeled as belonging to one of two classes, an SVM training algorithm builds a model that assigns new examples into one class or the other. Although a detailed explanation of SVMs is beyond the scope of this work, we refer interested readers to (Cortes and Vapnik, 1995) for foundational knowledge on SVMs and their mathematical formulations.

The training procedure for our models involved using KyTea's `train-kytea` command with the `-full` option to learn model parameters from our differently-sized datasets. These datasets, comprising varying numbers of kaiju-related Wikipedia articles, were created by processing said articles to extract the first two characters of each word, forming the basis for our autocompletion tasks. By training on these diverse datasets, we aimed to capture the linguistic diversity and complexity of kaiju-related language. More information about the data used can be found in the next section.

For testing, each trained model was used to make predictions on a processed version of the KINGKONG dataset (also described in the next section), which similarly contained only the first two characters of each word from the original article. The predictions were then compared against the ground truth values from the KINGKONG dataset. This comparison allowed us to evaluate the effectiveness of our models in accurately completing kaiju-related texts.

We employed two metrics to assess the performance of our autocompletion models: the unknown token ratio (U_{ratio}) and the Levenshtein ratio (L_{ratio}).

U_{ratio} is defined as the ratio between the number of unknown tokens in the input string (tokens not present in the respective training corpus) and the total number of tokens in the ground truth file. Formally, this is given by:

$$U_{ratio} = \frac{\text{\# of unknown tokens}}{\text{\# of tokens in ground truth}} \tag{1}$$

L_{ratio} is the ratio of the Levenshtein (edit) distance from the predicted string to the ground truth and the number of characters in the predicted string. This metric thus provides a measure of how much effort a potential user would need to put into correcting the autocompleted text. It is calculated as follows:

$$L_{ratio} = \frac{\text{Levenshtein distance}}{\text{\# of characters in predicted string}} \tag{2}$$

By analyzing these metrics, we gained insights into the robustness and accuracy of our autocompletion models. A lower U_{ratio} indicates fewer unknown tokens and thus better coverage of the language model, while a lower L_{ratio} signifies less effort required for correction, pointing to higher accuracy in autocompletion.

In the following section, we detail the creation of our kaiju corpora (for training) and KINGKONG dataset (for evaluation), together with their specifications.

3 LANGUAGE RESOURCE

To create our training datasets, we programmed a web-scraping tool to download the English

Wikipedia articles of the 9 most iconic Japanese kaiju according to Screen Rant (Draven, 2021). The kaiju include well-known figures such as Godzilla, Mothra, and others. After retrieving the articles, we processed them by eliminating special characters and converting them into a fully tagged form suitable for the `-full` training option of KyTea.

Each token in our datasets is composed of the first two characters (or one if the word/symbol has less than two characters), and its tag is the complete word or symbol. For example, consider the original phrase in an article: "Godzilla is a monster". The tagged form would be:

```
Go/Godzilla is/is a/a mo/monster
```

We created nine different training corpora, each with a unique size to investigate the dependence of model performance on training data size. The number at the beginning of each dataset name indicates how many Wikipedia articles the dataset contains. For instance, the 1KAIJU corpus contains text retrieved from 1 Wikipedia kaiju page, the 2KAIJU corpus contains text from 2 Wikipedia kaiju pages, and so on. Each Wikipedia page corresponds to the titular kaiju character, so the number is also equivalent to how many kaiju are represented in the dataset. The specification of each dataset can be found on table 1

Table 1: Specifications of the Kaiju Corpora

Name	# Kaiju	Tokens	Source
1KAIJU	1	4240	Wikipedia
2KAIJU	2	8369	Wikipedia
3KAIJU	3	12824	Wikipedia
4KAIJU	4	16695	Wikipedia
5KAIJU	5	17603	Wikipedia
6KAIJU	6	18477	Wikipedia
7KAIJU	7	19639	Wikipedia
8KAIJU	8	20410	Wikipedia
9KAIJU	9	23332	Wikipedia
KINGKONG	1	7152	Wikipedia

Each dataset contains varying numbers of tokens, which allows us to analyze how the performance of our models scales with the amount of training data. Note that, the number of total tokens does not necessarily correlate directly with the number of unique tokens, as many words appear multiple times within a single article or across different articles.

In the following sections, we will detail the results of our experiments.

4 EXPERIMENTAL EVALUATION

We trained L1-regularized and L2-regularized SVM models on all of the KAIJU datasets, resulting in a total of 18 models. The predictions were then evaluated on the KINGKONG dataset using the metrics detailed in the "METHOD" section. The results are displayed in Figure 1.

As seen in Figure 1, both the Levenshtein distance (L_{ratio}) and the ratio of unknown words (U_{ratio}) rapidly decrease as the size of the training dataset increases, and then stabilize after the inclusion of the 4th kaiju. This suggests that using more than 4 kaiju for training becomes redundant in the context of a dialectical battle with King Kong, reaffirming that SVM models perform well with relatively small training datasets.

Additionally, the L2-regularized SVM models consistently yield better accuracy, indicated by a smaller L_{ratio}, across all training sets compared to the L1-regularized models. This was expected, as L2 regularization tends to produce models with better generalization capabilities. However, the L1-regularized models are likely more compact, which raises questions about which model type would be more suitable for specific applications. This presents an interesting direction for future research.

Moreover, while it is expected that U_{ratio} should always decrease as the training dataset size increases, this trend is not perfectly observed in our results. The cause of this inconsistency has not yet been identified, indicating that further investigation is needed in future work to understand and resolve this anomaly.

5 CONCLUSION

In this study, we have demonstrated the feasibility and effectiveness of using SVM-based autocompletion models to enhance the creation speed for kaiju-related content. Our experiments reveal that both L1-regularized and L2-regularized models can significantly improve script completions for kaiju narratives, with L2-regularized models showing superior accuracy (decreased L_{ratio} across different training set sizes. Notably, our results suggest that training on a dataset containing texts from as few as four kaiju is sufficient to achieve robust performance in completing the King Kong Wikipedia article. This indicates that our approach is efficient

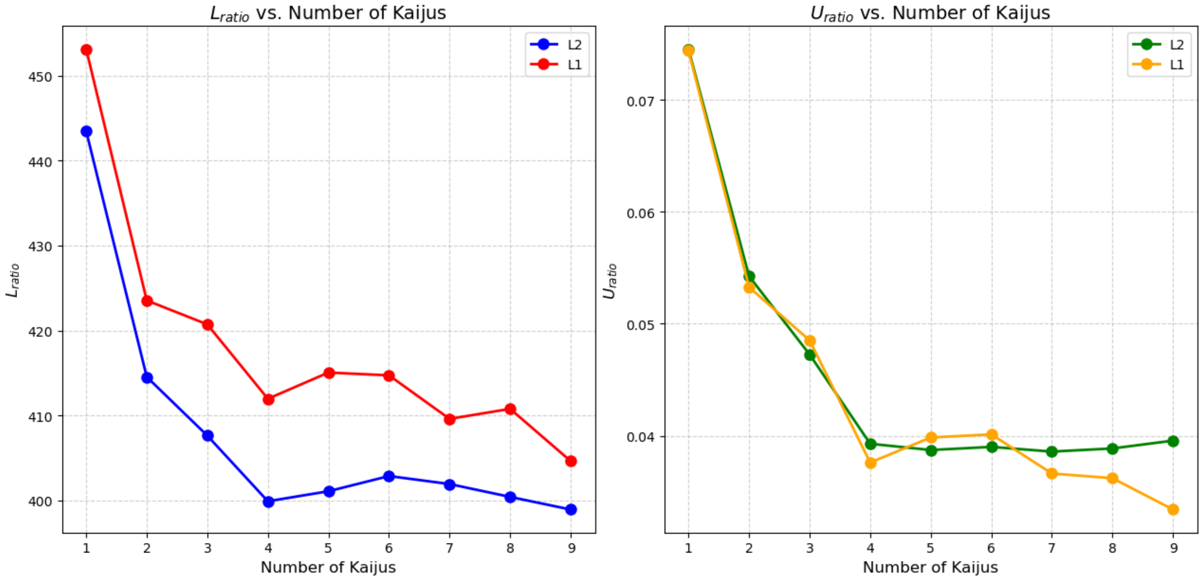

Figure 1: Performance comparison of L1-regularized and L2-regularized SVM models on the KINGKONG dataset. Note how both the Levenshtein distance and the ratio of unknown words rapidly decrease as the size of the training dataset increases, and then stabilize after the 4th kaiju.

and well-suited for niche applications where large datasets are not available.

While our findings are promising, there are areas that require further investigation. The unexpected fluctuations in U_{ratio} indicate that there may be underlying factors affecting the model's inner workings that we have yet to understand. Future work should aim to identify and address these anomalies to further refine the models.

Additionally, our work opens up several interesting research directions. For instance, exploring the trade-offs between model compactness and accuracy could provide insights into which type of SVM model—L1 or L2 regularized—is more appropriate for specific use cases. Exploring other types of models - such as logistic regression, or LLMs with in-context learning - would also be interesting. Furthermore, extending our approach to other niche domains could validate the generalizability of our methods.

In conclusion, our work provides a novel contribution to the field of kaiju content creation, offering a tool that supports the creative efforts of enthusiasts. By focusing on the linguistic prowess of kaiju articles, we have added a new dimension to the exploration of these iconic monsters, enabling the enrichment of narratives that continue to captivate audiences worldwide.

References

Mariette Awad and Rahul Khanna. 2015. *Support Vector Machines for Classification*, pages 39–66. Apress, Berkeley, CA.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Derek Draven. 2021. Godzilla: The 15 most iconic kaiju, ranked.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Ishirō Honda. 1962. King kong vs. godzilla.

Ishirō Honda. 1967. King kong escapes.

Graham Neubig, Mizuki Nakata, and Shinsuke Mori. 2011. Kytea: A toolkit for tokenization and text analysis. *Proceedings of the ACL 2011 System Demonstrations*, pages 19–24.

Adam Wingard. 2021. Godzilla vs. kong.

Adam Wingard. 2024. Godzilla x kong: The new empire.