Predicting SCP Entities' Object Class by their wiki contents

Iñigo Artolozaga de Ariño
i.artolozagadearino@student.utwente.nl

Mohamed Reda Benkhelifa
m.r.benkhelifa@student.utwente.nl

Group 43
Natural Language Processing Project
University of Twente
October 2021

Abstract

The SCP Foundation is a fictional secret organization that handles entities threatening to human life, and is documented by the collaborative writing wiki project of the same name. This wiki consists of pages for different threatening entities that the foundation handles classified by their danger level. In this project, we try to predict the danger level (called Object Class officially) just with the wiki page content of an SCP entity's wiki page. We scrape data from the SCP Wikia website. We process the aforementioned sections of the text with different tokenizers and train different Naive Bayes-based models with it.

1 Introduction

1.1 Context

The SCP Foundation is a fictional secret organization that handles entities threatening to human life, and is documented by the collaborative writing wiki project of the same name. This popular piece of media is often categorized in the horror category due to the nature of the threats described in it. The SCP Foundation is a completely digital phenomenon with which fans engage from all over the world.

The SCP Wikia page, which is an official and extremely curated repository, contains an encyclopedia of all the threatening entities that the SCP Foundation has had to deal with. The wiki entries are presented in a format that reminds the reader of a declassified military document, lending a sense of legitimacy and mystery to its contents. As it can be seen in the example wiki entry in the Appendix, a description of their appearance and nature, specific containment instructions and their Object Class are common sections in this website. The Object Class, according to the wiki, "serves as a rough indicator for how difficult an object is to contain" and "is for the purpose of identifying containment needs, research priority, budgeting, and other considerations". In layman's terms, the more highly ranked a threat's Object Class is, the more effort (in terms of resources) is needed to contain it. The three main Object Classes are Safe, Euclid and Keter. Safe entities are easily and safely contained, Euclid class entities are more difficult to secure and often less reliably contained, and lastly, Keter class threats are exceedingly difficult to contain, with containment procedures usually being extensive and complex.

1.2 Research Question

In this paper, we want to:

- Ascertain the level of descriptiveness that different Wikia sections have with regard to the Object Class.
- Test the performance of different classification methods with the dataset composed of SCP Wikia entries.

To answer these questions, we will apply different methods seen in the Natural Language Processing course to classify data from the SCP Wiki. We will then compare the obtained results to determine the best method of classification for this specific dataset.

2 Related Work

No quantitative research exists on the SCP Foundation and its wiki, but we found some papers that use similar analysis methods than us:

There are different ways to do a multi-class text classification. The "common text classification methods are Naive Bayes Classifier for Multinomial Models, Linear Support Vector Machines, Logistic Regression, Word2vec, Logistic Regression, Doc2Vec and Logistic Regression, BOW with Keras" [4]. Waqas Arshad et al. (2021)[4] applied those methods on a Stack Overflow question dataset and compared the obtained accuracies in order to estimate which classification method gives the best results. They sought to predict which tag matched to the post by analysing the post and the various comments attached to it. They concluded that Doc2Vec was the best method for that dataset.

Among other existing works, Yong Wang et al. (2003) [7] have done something similar to us, that is to say, classify data retrieved from the web using naive Bayes method. They extracted NLP components from the HTML files using webdoc and parsed the documents to isolate essential components and then separated documents for the training and testing phase. They compared the precision, the recall and the F-measure by applying the Multinomial Naive Bayes model and Multi-variate Bernouilli Model but also applied different smoothing methods. We discuss smoothing in the Discussion section, but it was not a central part of our work.

According to Guo Qiang (2010), it is possible to improve the performances of the text classification performance of the Naive Bayes Multinomial model[5], which is one of the three models that we will use in our experimentation. The author claims that ”the multinomial Naive Bayes model treats each occurrence of a word in a document independently of any other occurrence of the same word. However, multiple occurrences of the same word in a document are not independent.” [5]. A way they propose to improve this is to change the way of counting the words in the document. According to the results obtained, applying this method substantially increases the performance of the model. However, due to the encyclopedia-like nature of our dataset, we are skeptical if most of the assumptions made in this paper apply to our case. In this project we test its precision.

3 Data

For this project, we decided to exclusively use data from the SCP Wikia page[2]. This wiki-style website contains an entry for each SCP entity, so it is perfect for our use case. There are, at the time of the execution of this project, almost 7000 SCP entities recorded in this repository. They are organized by series of 1000 SCP entities each, which have been added over the years ever since the creation of the page. As more are created, they are included in the current series until it is full, in which case a new Series is created. The entities are identifiable by a unique ID assigned to them. This fact made the data wrangling process easier for us. For our dataset, we used all 7 series of SCP entities, so SCP-001 to SCP-6999. However, not all of these IDs are occupied by an entity at the time of writing this report, so we prepared for it in our scraping methods.

To build our dataset, we first had to scrape all the necessary data from the wiki. Luckily, we found an open source tool to do so called SCPScraper[3], which had basic functionality to get the HTML page content of a desired entity, taking its ID as an argument. After fetching the HTML source code, we looked at its contents and extracted the sections which were relevant for us. In this case, we extracted the Object Class, the Description and the Special Containment Procedures sections off of the source HTML by looking for their specific HTML tags. For the Object Class, however, we ended up looking at the wiki entry's tags, which we found out is a more consistent way of finding it than looking at the text. Lastly, we built a json data structure and stored all the extracted sections under the same ID. We exported this information as a json file (attached as ”scps.json”) to import it in further stages of the project.

This process was not trivial, since the pages were not as homogeneously structured as expected, especially in terms of source code. Our methods for extracting the data can be found in the attached notebook named ”NLP_Project_Scraping.ipynb”.

4 Methods

With the objective of answering our Research Questions, we decided to tinker with all the steps of performing a Naive Bayes classification. For our case, this meant using the following as our dependent variables in our evaluation:

- Dataset: Is the Description of the wiki page section enough to guess the Object Class, or is it necessary to also use the Special Containment Procedures section to be informative enough?
- Stemming: Is stemming effective for this dataset at all?
- Vectorization: What vectorizer performs better with this particular dataset?
- Naive Bayes Model: What Naive Bayes implementation is more effective in classifying SCP data?

To obtain a train and a test dataset, we used a method out of the sklearn model selection library, entitled StratifiedKFold[1]. We initially intended to use the ”KFold” method from the same library, but we later discovered, as it can be seen in Figure 1, that the dataset is quite unbalanced. Therefore, we considered it opportune to use a method that could create training sets with as balanced as possible a distribution of object classes.

Object Class	Amount of Entities
Safe	1814
Euclid	2093
Keter	968

Figure 1: Amount of entities for each Object Class in the dataset.

This is also the main reason why we decided to experiment with, among others, the Complement Naive Bayes model. This model, ideated by Rennie et al. (2003), has the objective of ”tackling the poor assumptions made by Naive Bayes text classifiers”[6], especially the Multinomial Naive Bayes model. This model shows promise for our implementation.

5 Results

With all the variables described in the Method section taken into consideration, we tested a total of 36 combinations, which can be seen in Figures 2, 3 and 4. For the code used in this section, please refer to the attached notebook entitled ”NLP_Project_Modelling.ipynb”.

	CountVectorizer		TfidfVectorizer	
	Stem	No	Stem	No
Descriptions	41.97	43.05	42.40	43.42
Sp. Procedures	40.92	42.83	39.63	42.44
Both	41.21	42.46	41.91	42.22

Average: 42.04%

Figure 2: Precision of different preprocessing configurations with a Gaussian Naive Bayes Model (%).

	CountVectorizer		TfidfVectorizer	
	Stem	No	Stem	No
Descriptions	52.51	52.90	47.55	47.91
Sp. Procedures	56.49	55.96	53.07	53.55
Both	42.46	56.53	48.12	48.59

Average: 51.30%

Figure 3: Precision of different preprocessing configurations with a Multinomial Naive Bayes Model (%).

	CountVectorizer		TfidfVectorizer	
	Stem	No	Stem	No
Descriptions	51.77	51.94	50.21	50.87
Sp. Procedures	55.03	54.67	55.53	55.44
Both	54.54	55.18	52.21	52.62

Average: 53.33%

Figure 4: Precision of different preprocessing configurations with a Complement Naive Bayes Model (%).

Some observations can be made from the results of the experimentation process:

- The ”Special Containment Procedures” section (labeled in the tables as ”Sp. Procedures”) provides more information about the entity's Object Class than its ”Description” section. This is most likely due to the superior descriptiveness of the actual danger that the entity poses present in the ”Special Containment Procedures” section.

- Combining the ”Descriptions” and ”Special Containment Procedures” datasets together does not have a substantial impact on the precision of the classification.

- Stemming does not seem to have a substantial impact on the precision of the classification.

- Strangely, using a CountVectorizer leads to marginally better scores when compared with Tf/idf Vectorizer implementations.

- The dataset is very heterogeneous, both in terms of category distribution and in terms of older entities versus newer entities. See Discussion section.

6 Discussion

While the results are satisfactory, there are multiple other variables that we did not take into consideration when carrying out the experimentation due to time and computational power constraints, and aspects of the implementation that did not behave as we initially expected:

6.1 Dataset

Although the dataset extracted from the SCP Wiki is extensive, our scraping procedures were conservative with regard to the structure of the page contents. If an SCP page deviated slightly from the usual structure, we did not use it for the dataset. This resulted in us only using 70% of the wiki entries for the dataset. With a more forgiving and sophisticated scraping method we could have obtained a bigger dataset and trained the model more extensively. We decided to try the same tests with a smaller sample, this time consisting of only the first 4000 SCP entity entries, as opposed to the original 7000. Figure 5 shows that, surprisingly, the results for this reduced dataset are substantially better.

Model	Gaussian	Multinomial	Complement
Avg (%)	46.08	58.48	60.28

Figure 5: Average of precision of different models with a dataset consisting only of the first 4 Series of SCP entities (N=4000).

This leads to the theory that the newer SCP entries are much more abstract and introduce noise in our dataset. With more time, it would have been interesting to explore this theory more in depth.

6.2 N-grams

Both the vectorizers that we use for our experimentation accept a parameter called ”ngram_range”, which specifies ”the range of n-values for different word n-grams or char n-grams to be extracted”[1]. Even though we could have introduced an n-gram range to process in the vectorizer implementation, we used only unigrams in the vectorization of the datapoints. We did this because using higher-dimensional n-grams was too demanding on our hardware and because it did not substantially improve the performance of any of the three tested models.

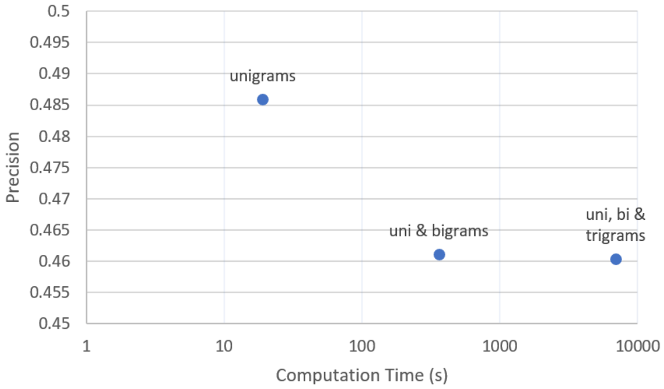


Figure 6: Counting n-grams is not worth it in terms of precision or computation times.

6.3 Vectorizers

Although during the course we learned that Tf/idf-based vectorizers are more sophisticated and offer more accurate results than count-based vectorizers, this was consistently not the case for our dataset. We had 18 different combinations for each type of vectorizer, with no stopwords and with 5 instances of cross validation per combination, and the CountVectorizer is more precise in two thirds of the cases. One possible explanation for this is that common terms are actually relevant for the classification of instances of this dataset, but the Tf/idf model is neutralizing them because of their high idf coefficient.

6.4 Smoothing

We did not include explicit indications for the smoothing when creating our Naive Bayes models. According to the Scikit-learn website, the default value for smoothing in all the Naive Bayes models we used is 1, which is automatically assigned if the argument is not explicitly declared. However, as it can be appreciated in Figure 7, the default setting may not necessarily be the optimal one.

With more time and greater computational power, it would have been interesting to take this variable into consideration in our research.

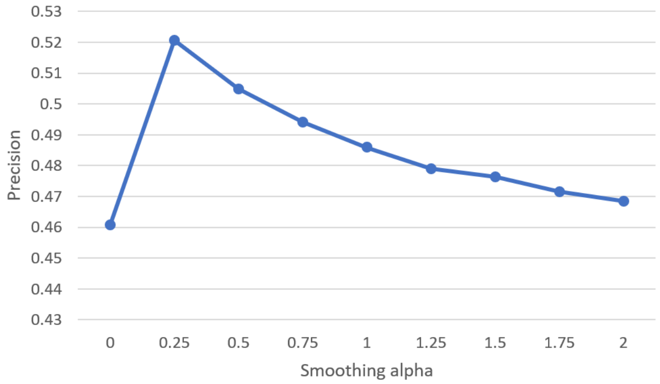


Figure 7: Smoothing strength affects the precision.

7 Conclusion

We built a dataset by scraping sections off of SCP Wikia entries. We then tried to predict their Object Class (danger level) based on different sections. We also experimented with different classification model implementations. Our results show that the ”Special Containment Procedures” section of the entries is the one that leads to the most precision. The combination of the ”Description” and the ”Special Containment Procedures” sections achieves lower accuracies, even though it increases the size of the training set.

The Multinomial Naive Bayes model and its subsequent adaptation, the Complement Naive Bayes model, are the most effective in this particular task and dataset. In particular, the Complement-based model compensates some of the severe assumptions made by its predecessor and works better for an imbalanced dataset such as the one we worked with, by preventing overfitting the model in favor of the most common Object Class. However, we consider that the advantage of this model over the Multinomial one was partly reduced by our usage of stratified cross validation, which allowed us to train the models on more equally distributed samples.

On top of this, we found that stemming the corpus did not have a tangible impact on the results. Additionally, we were surprised to found that, for this dataset, a simple count-based vectorizer was more accurate than a more sophisticated tf/idf-based one.

References

[1] Scikit-learn, https://scikit-learn.org/stable/modules/.

[2] Scp foundation wikia, https://scp-wiki.wikidot.com.

[3] Scpscraper, github - https://github.com/jaonhax/scpscraper.

[4] W. Arshad, M. Ali, M. Mumtaz Ali, A. Javed, and S. Hussain. Multi-class text classification: Model comparison and selection. In *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–5, 2021.

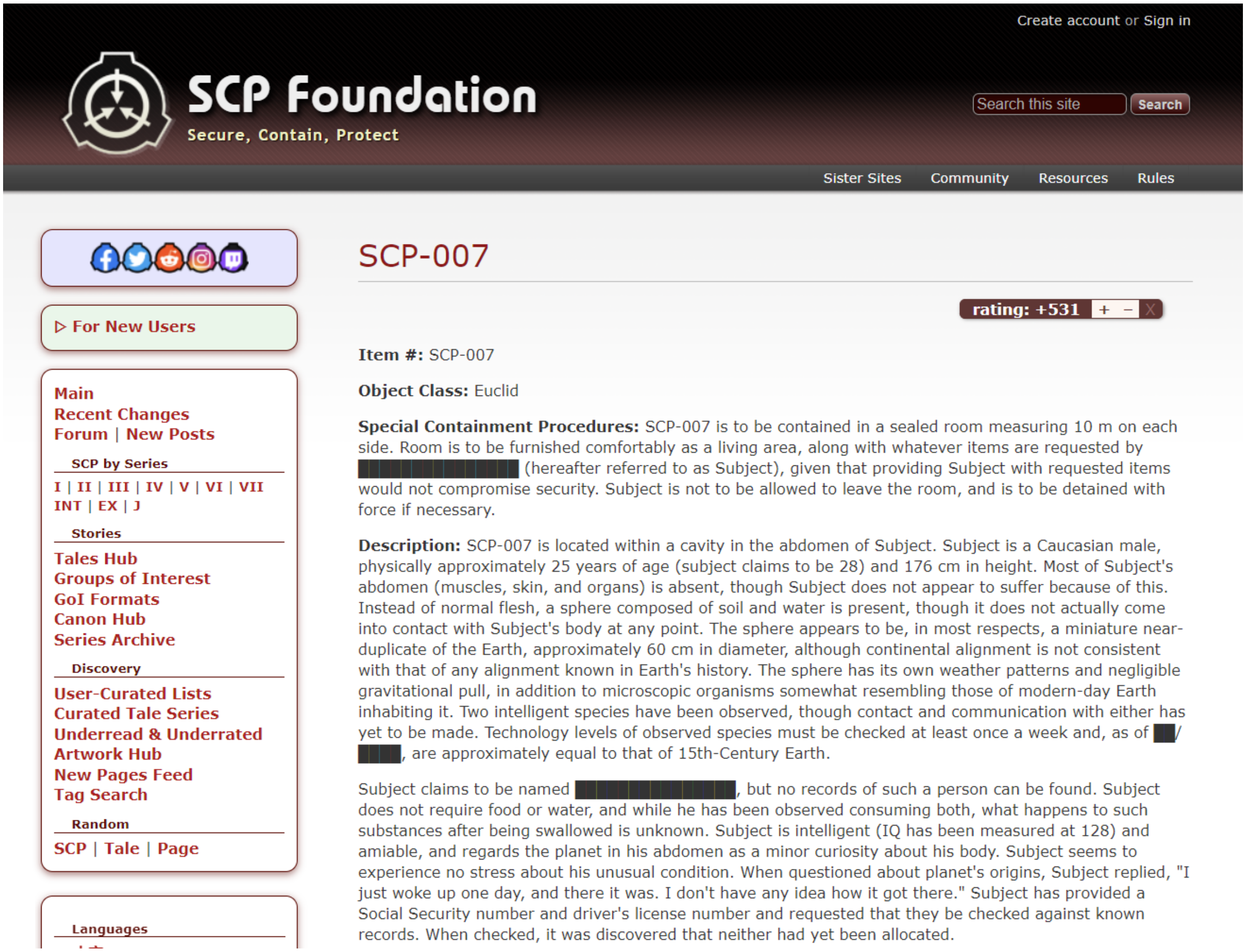
[5] G. Qiang. An effective algorithm for improving the performance of naive bayes for text classification. In *2010 Second International Conference on Computer Research and Development*, pages 699–701, 2010.

[6] J. Rennie, L. Shih, J. Teevan, and D. Karger. Tackling the poor assumptions of naive bayes text classifiers. 01 2003.

[7] Y. Wang, J. Hodges, and B. Tang. Classification of web documents using a naive bayes method. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 560–564, 2003.

8 Appendix

8.1 Example of an SCP Wiki page



SCP-007

Item #: SCP-007

Object Class: Euclid

Special Containment Procedures: SCP-007 is to be contained in a sealed room measuring 10 m on each side. Room is to be furnished comfortably as a living area, along with whatever items are requested by [redacted] (hereafter referred to as Subject), given that providing Subject with requested items would not compromise security. Subject is not to be allowed to leave the room, and is to be detained with force if necessary.

Description: SCP-007 is located within a cavity in the abdomen of Subject. Subject is a Caucasian male, physically approximately 25 years of age (subject claims to be 28) and 176 cm in height. Most of Subject's abdomen (muscles, skin, and organs) is absent, though Subject does not appear to suffer because of this. Instead of normal flesh, a sphere composed of soil and water is present, though it does not actually come into contact with Subject's body at any point. The sphere appears to be, in most respects, a miniature near-duplicate of the Earth, approximately 60 cm in diameter, although continental alignment is not consistent with that of any alignment known in Earth's history. The sphere has its own weather patterns and negligible gravitational pull, in addition to microscopic organisms somewhat resembling those of modern-day Earth inhabiting it. Two intelligent species have been observed, though contact and communication with either has yet to be made. Technology levels of observed species must be checked at least once a week and, as of [redacted]/[redacted], are approximately equal to that of 15th-Century Earth.

Subject claims to be named [redacted], but no records of such a person can be found. Subject does not require food or water, and while he has been observed consuming both, what happens to such substances after being swallowed is unknown. Subject is intelligent (IQ has been measured at 128) and amiable, and regards the planet in his abdomen as a minor curiosity about his body. Subject seems to experience no stress about his unusual condition. When questioned about planet's origins, Subject replied, "I just woke up one day, and there it was. I don't have any idea how it got there." Subject has provided a Social Security number and driver's license number and requested that they be checked against known records. When checked, it was discovered that neither had yet been allocated.