ISSN 1870-4069

Anime Success Prediction Based on Synopsis Using Traditional Classifiers

Jesús Armenta-Segura, Grigori Sidorov

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{jarmentas2022, sidorov}@cic.ipn.mx

Abstract. For predicting the success of an anime in its early stages of development, a baseline is proposed in this paper, based on the synopsis of its plot. AniSyn7 is presented, which is a corpus consisting of $6,928$ anime synopsis with binary labels of successful/unsuccesful. The corpus was explored by vectorizing the synopsis using n-grams and dependence trees, so three traditional machine learning classifiers (Support Vector Machine, Gaussian Naive Bayes, and Logistic Regression) can be employed in order to study correlation between synopsis and success.

Keywords: Anime success prediction, entertainment success prediction, machine learning, natural languaje processing.

1 Introduction

In recent years, the anime industry has become highly profitable. For example, Jujutsu Kaisen, the most lucrative anime of 2022,[1] generated $76 million in Japan alone. However, failure is equally punished, as demonstrated by the anime movie Final Fantasy: The Spirits Within, which lost more than $50 million.[2] Given that high risk on the investment, being able to predicting the success of an anime in early stages of development is crucial for the industry.

The success of an anime depends on several factors, including plot quality, animation, voice acting, soundtrack, marketing, among others. However, in the initial stages of development, plot features are the most accessible and cost-effective to assess as it happens with movie productions [7]. Hence, inspecting the correlation between plot quality and success is a prudent starting point for research.

Plots can be summarized through its synopsis, which can also be marketed as a standalone product, as it happens in Hollywood [5]. Hence, reducing the plot to its synopsis is a very effective and low-cost way to start this inspection. About studying correlation between variables, in recent times, machine learning classification techniques have emerged as powerful tools for tackle that issue, because their ability to

[1] According to Oricon's success rating: www.oricon.co.jp/special/61353/5/ An english explanation can be find in erzat.blog/oricons-yearly-manga-sales-ranking-2022/

[2] www.boxofficemojo.com/release/rl3008595457/

		Predicted by the method	
		Successful	**Unsuccessful**
Real	**Successful**	True Positives (TP)	False Negatives (FN)
	Unsuccessful	False Positives (FP)	True Negatives (TN)

Fig. 1. True is for correctly labeled and False for wrong labeled. In this case, positive means successful and negative means unsuccessful.

learn and identify patterns and relationships within corpora. Support Vector Machine (SVM), Naive Bayes (NB), and Logistic Regression (LogReg) are three of the most prominent classifiers, and had proven to be useful by establishing performance baselines in other text related tasks [8, 13, 14, 19, 12, 6, 17, 18].

Hence, inspecting the correlation between plot and success through them will conform a suitable baseline for the anime success prediction task. In this paper we present AniSyn7: a balanced corpus with $6,928$ anime synopsis scrapped from MyAnimeList (MAL)[3]. They are associated with a binary golden label based in MAL's weighted score.

We explore this corpus with the three machine learning classifiers mentioned above, and we gain valuable insights about the correlation between synopsis and success, through their confusion matrices. Our most successful approach was Bag of Words + Bigrams + Trigrams + Character Trigrams classified with Support Vector Machine and Logistic Regresion (BBTC+SVM and BBTC+LogReg), which both achieved an F1 score of 0.55.

2 Related Work

Several work has been done in the most general task of entertainment success prediction through natural language processing (NLP) and machine/deep learning techniques. For instance, in [7] the authors used deep learning for predicting movie success. In [8, 9] the authors tackled the book success prediction task:

In [9], they used a multitask setting to predict both success and genre, which led to a significant improvement in performance with respect to their state of the art; whereas in [8] they used neural networks and transformers with an original vectorization based on the emotional lexicon of the text, which yield an improvement of the results from [9].

Focusing on anime success, there are various works on the closely related task of recommender systems design [4, 11, 16]. The most closely related work to the success prediction task is [1], where the authors studied the correlation between reviews and success using sentiment analysis in the reviews. However, they utilized a very unbalanced dataset with less than 5% of negative reviews, so they used data augmentation techniques.

[3] myanimelist.net/

[4] Image source: https://scikit-learn.org/stable/modules/cross_validation.html.

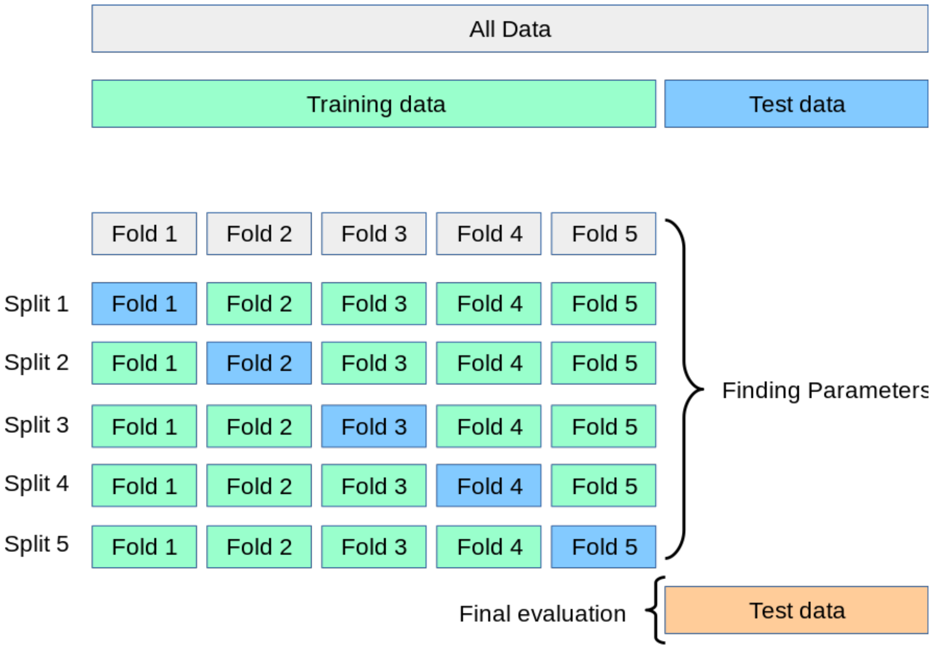


Fig. 2. In K-fold cross validation, the MLC must to be trained K times, each time taking a different fold as test set (in dark blue). At the end, the model is checked with the test data[4].

3 Methods

3.1 Vectorization

Machine learning classifiers (MLC) must to be feed with a vectorial representation of the data. In this paper we focus in three vectorizations: two based in N-grams and one based in dependence trees.

***N*-grams:** A token is defined as a fundamental unit of text, such as a word or a single character. An N-gram is defined as a contiguous sequence of N tokens. By counting the frequencies of appearance of N-grams, it is possible to vectorize a text, as follows:

1. We collect all the N-grams from all synopsis in the entire dataset, and create a vocabulary consisting of all non-repeated N-grams.
2. Finally, we represent each text in the dataset as a vector, where the i-th entry corresponds to the frequency of the i-th N-gram in the vocabulary, who can be zero.

This approach captures some lexical information and have prove to be useful in tackling NLP tasks as sentiment analysis, authorship attribution and fake news spreaders detection [2, 3, 10, 15, 12]. Usually, this vectorizations also comes along with lemmatization and stopword removal, so we do the same.

In this paper we used Bag of Words (BoW), where $N = 1$ as well as trigrams, where $N = 3$, with tokens defined as words. Besides BoW alone, we also applied various N-grams vectorizations together, at the same time:

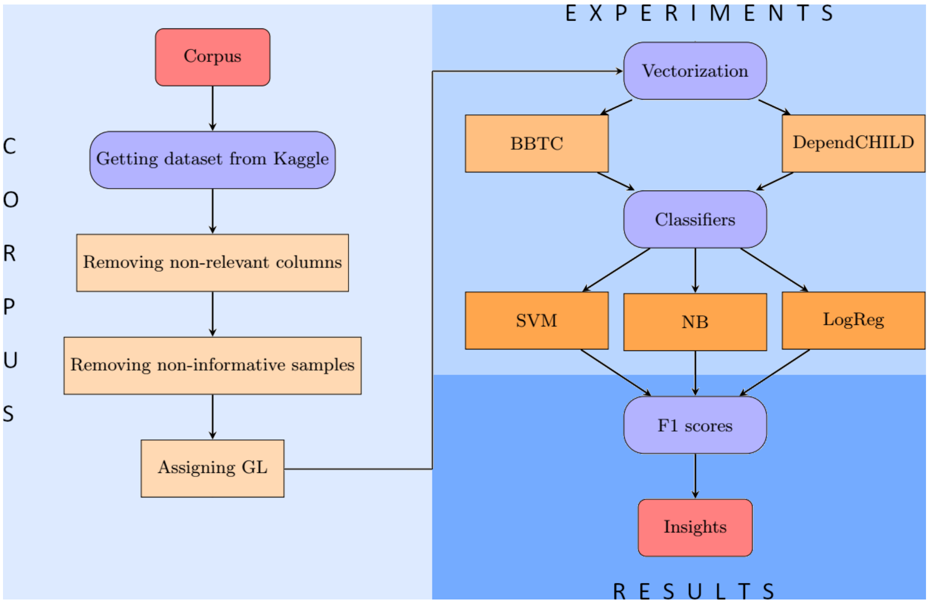


Fig. 3. Workflow for the anime success prediction task.

BBTC is an abbreviation for "Bag of Words, bigrams, trigrams, and character trigrams", which are $1, 2, 3$-grams respectively. In the first three cases, the tokens are defined as words, whereas in the fourth case the tokens are defined as individual characters. We used the Stanza python library for lemmatizing and removing stopwords, as well of the re package to remove special characters through regex.

Dependency trees: They are a way to represent the syntactical structure of a sentence by breaking it down into a hierarchical structure of grammatical dependencies between words. In a dependency tree, each word in a sentence is represented as a node, and the syntactical relationships between them corresponds to directed edges.

A naive baseline to take advantage of this syntactical representation is by counting the children of every node in a dependency tree (DependCHILD). This allows to vectorize a text, as follows:

1. We get the dependency tree of a synopsis, using Spacy python library.
2. We create a vector whose i-th entry corresponds to the amount of children of the i-th word.
3. We ensure uniform vector length by adding zeroes as needed until we reach the size of the largest vector.

As long as we know, this naive approach has never been used before in other NLP tasks. However, we admit that is extremely rudimentary and should be used only for stating baselines purposes.

Table 1. Examples of animes whose synopsis are not informative.

Anime Title	Synopsis
Poo Poo kids	A comedy series about farting children
Kabukichou Sherlock OVA	A fun collection of six short side stories depicting the past and present of Sherlock, Watson, Moriarty, and the rest of the Row House as they go about their everyday lives in their home of Kabukicho
Virtual-san wa Miteiru	Virtual-san wa Miteiru is an anime that brings Virtual YouTubers to life. The term Virtual YouTuber, or VTuber, refers to (...) Virtual-san wa Miteiru is a one-of-a-kind comedy certain to bring delight to any fan of Virtual YouTubers!

3.2 Classifiers

SVM: It is a supervised MLC that approximates the best separation between data. In this case, we ask SVM to finds the best hyperplane (line, plane, volume, etc.) whose defined semi-spaces divided successful animes from unsuccesful. The performance of SVM depends on how well the data can be separated by such hyperplane.

If several successful animes result to be nearby to several unsuccessful animes, SVM may not be able to make a proper separation. Since the vectors are defined in terms of selected features, this failure of SVM means that the phenomenom is not properly characterized by those features.

NB: Is a probabilistic MLC that assigns a label through a random variable who is calibrated using a dataset with binary golden labels. In this case, Naive Bayes computes the probability of an anime being successful, given the vectorial representation of its synopsis, and viceversa. For that computation, it uses the generalized Bayes theorem applied to every feature f_i, assuming that their appearance in the synopsis are independent of each other[5]:

$$P(\text{Success}|f_1,\ldots,f_n) = \frac{P(f_1,\ldots,f_n|\text{Success}) \times P(\text{Success})}{P(f_1,\ldots,f_n)}. \tag{1}$$

The performance of NB strongly depends on the presence of a feature in every labeled data. If a synopsis contains a majority of features that NB assigned a higher probability of being related with success, then that synopsis will be labeled as successful.

Wrong labeling can happens if either several features of the synopsis has probabilities very near to $1/2$, which means that every feature has similar presence in both labelings, and hence there is no statistical correlation. In this work we use Gaussian Naive Bayes, who assumes a normal distribution for the random variable.

Logistic Regression (LogReg): It is a supervised MLC similar to linear regression, but for binary predictions (successful/unsuccessful). The algorithm uses a sigmoid function to map the vectorized synopsis to a probability value.

[5] Since that rarely happens in practice, this classifier is considered as naive.

Table 2. Discarding of underrepresented genres.

Genre	Description	Merged with	Justification
Unknown	No genre	eliminated	For noise reasons.
Gourmet	About gastronomy	eliminated	It is actually a theme.
Award Winning	Won a prestigious award	eliminated	Feature beyond of plot.
Avant Garde	Experimental animes	eliminated	Although narrative is distinct, contents are not.
Mystery/Suspense	About riddling mysteries	eliminated	The most painful loss since they are well-stablished genres beyond anime.
Boys/Girls love	Softporn romance	Romance	Softporn is not considered Hentai (proof: Fairy Tail is not considered Hentai).
Erotica	Explicit erotism	Hentai	Although Hentai (変態) means pervert, we decided to take it as a synonym of porn.
Sports	About sports	Action	Sports are full of action.
Horror	Frightening content	Supernatural	Usually horror is based in paranormal events (not always).

If the probability is greater than a certain threshold, the anime is predicted as successful, otherwise it is predicted as unsuccessful. The algorithm learns the coefficients of the input variables by minimizing the difference between the predicted probability and the actual output value in the training data.

3.3 Score Measures

$F1$-score is a popular measure for MLC correctness. It is calculated in base of how many successful/unsuccessful animes were correctly classified or not.

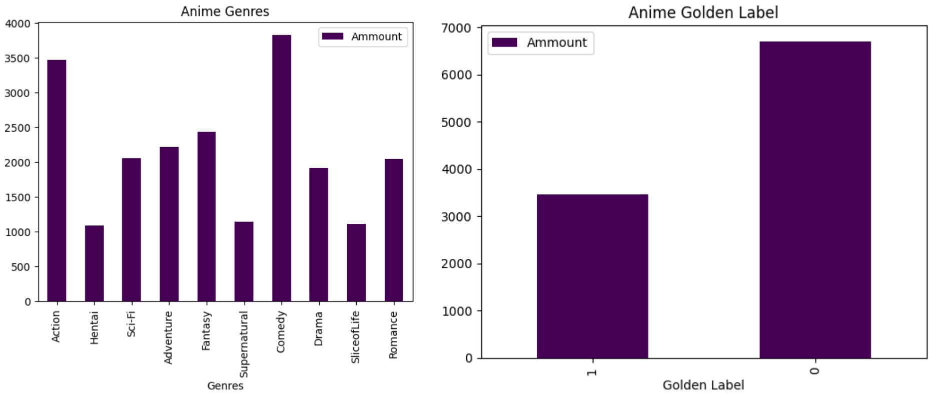
- **Confusion matrix:** It is a nice way to visualize the raw performance of a MLC. It is a 2×2 matrix, defined in Figure 1.
- **Accuracy:** Is the mean of all correct labeled samples by the MLC, within the corpus:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{2}$$

where TP are the true positives, TN the true negatives, FP the false positives and FN the false negatives.

- **Presicion:** Measures how effective was the successful-labeling of the MLC. It consider all animes labeled as successful and compares them with the only the correct labeled:

$$\text{Presicion} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3}$$



(a) (b)

Fig. 4. The dataset after preprocessing but before genre and golden label balance. Hentai is the less represented with $1,092$ samples.

- **Recall:** Measures how effective the MLC labeled successful animes. It consider all successful animes of the dataset and compares them with the correctly labeled ones:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4}$$

- **F1 score:** It is the harmonic mean of precision and recall. This is the most widespreaded score for binary classification:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Presicion} + \text{Recall}}. \tag{5}$$

3.4 K-fold Cross Validation

For training a MLC, the dataset must be splitted into a training data, used for calibrates the classifier, and the test data, used for checking the performance. K-fold cross validation consist in splitting the training data into K parts, and then to train the MLC taking one of that parts as a test set (Figure 2). Finally, we check the model by doing predictions in the test data.

4 Experiments

We created a suitable corpus, named AniSyn7, by processing an anime synopsis dataset, focusing it on the success prediction task. We vectorized them with BBTC and DependCHILD, and finally we applied SVM, NB, and LogReg for classify them. We gathered the confusion matrices, calculated accuracy and $F1$ scores, in order to study correlation between success and synopsis. See Figure 3 for the workflow.

Algorithm 1 Balancing the corpus.

```
procedure BALANCER(corpus in xlsc format)
    $\ell \leftarrow$ genre with less animes in the corpus
    $\#\ell \leftarrow$ ammount of animes with genre $\ell$
    $dik \leftarrow$ {genre name : #animes with that genre $- \#\ell$}

    $g, G \leftarrow$ Golden label with minimum/maximum ammount of animes
    $\#g, \#G \leftarrow$ Ammount of animes with gl g/G

    while $\#G - \#g > 0$ do
        Victim $\leftarrow$ An anime with golden label $G$
        if for all genre $\mu$ of Victim, dik$[\mu] > 0$ then
            Remove Victim from the corpus
```

4.1 AniSyn7 Corpus

Dataset The initial dataset was gathered from Kaggle[6], and consists in 21,460 anime series, movies and musical videos scrapped from MAL by the user Harits Fadlilah. This dataset includes genre, theme, demographics, year, format, and also includes the MAL's **weighted score** W, calculated with the follow formula[7]:

$$W = S\left(\frac{v}{v+m}\right) + C\left(\frac{m}{v+m}\right), \tag{6}$$

where S is the average score for the anime, v is the number of users who scored, m is the minimum number of scored users to get a score and C is the mean score across the entire Anime/Manga database.

Weighted score justification: A naive baseline for measuring anime success is the mean of all user scores, given by:

$$S = \frac{\text{Sum of all users scores of the anime}}{\text{Total ammount of users who scored the anime}}. \tag{7}$$

However, this can be biased by fake reviews, so MAL filters out votes from users who have not watched at least a fifth part of the anime. Also, they determined a minimum of $m = 50$ ratings for S to be statistically significant,[8] which also associates S with a weight, in terms of m, given by:

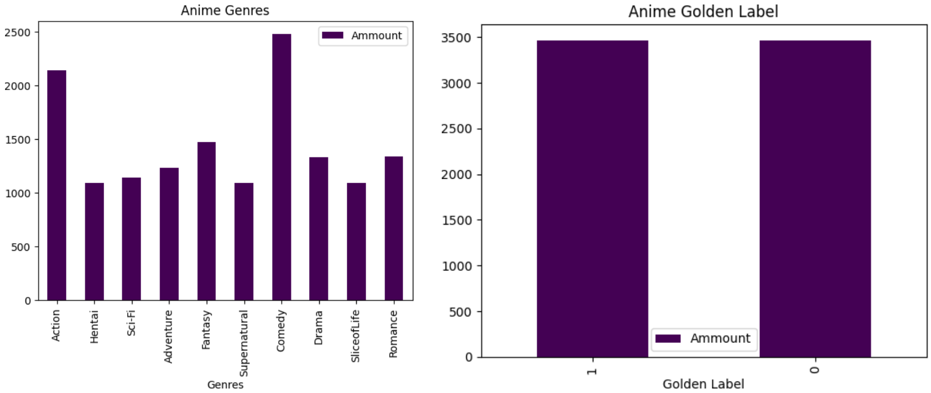
$$\text{Weight of } S \text{ in the score} = \left(\frac{v}{v+m}\right). \tag{8}$$

If v grows, $v/(v+m)$ tends to 1. When $v = 1$, is reached the minimum nonzero value $1/(m+1)$, who also is the statistical importance of a single review. For an anime with no human scores, MAL assigns a default score of:

[6] www.kaggle.com/datasets/harits/anime-database-2022

[7] myanimelist.net/info.php?go=topanime

[8] Value of m at least until 2023.



(a) (b)

Fig. 5. The corpus after genre and golden label balance. No Hentai was eliminated, and Comedy stayed as the most represented genre, with $2,478$ samples.

$$C = \frac{\text{Sum of all valid scores in the database}}{\text{Total ammount of valid scores in the database}}. \tag{9}$$

Who is actualized twice per day,[9] but is treated as a constant in every calculation. Analogous to S, the weight of C in the score is defined by:

$$\text{Weight of } C \text{ in the score} = \left(\frac{m}{v+m}\right). \tag{10}$$

If v grows, $m/(v+m)$ tends to 0. If $v = 0$, then C's weight is 1, which reflects the fact that MAL assigns C as a default score for an anime that has never been reviewed by any user.

Making the corpus: In order to focus on the anime success prediction task, we extracted only the follow relevant columns from the dataset: title, synopsis, score, genre, demographics, and theme. We were inspired by the multitask setting proposed in [9] who yielded better results for the book success prediction task, but we will not explicitly use it in this work. Instead, we implicitly incorporate genre information through balancing (Algorithm 1), leaving open the possibility of explicitly explore this approach in further work.

Preprocesing: We cleaned the dataset by eliminating noisy entries, according to the follow criteria:

1. **Non-informative synopsis:** There are some animes whose synopsis describes the product instead the plot, or is very ambiguous. In Table 1, are depicted three examples. We filter out such entries with a simple but effective criterion: we only included synopses with a minimum length of $1,000$ characters. While a few entries, such as Virtual-san wa Miteiru, may have slipped into the final corpus, they are not significant enough to introduce statistical noise.

[9] myanimelist.net/topanime.php

Table 3. F1 and accuracy scores for the synopsis-based anime success classification task. $\text{BBTC} + \text{LogReg}/\ \text{SVM}$ had the best performance.

Methods	TP	FP	TN	FN	Accuracy	F1
BBTC+LogReg	336	191	502	357	0.60	0.55
BBTC+SVM	342	208	485	351	0.59	0.55
DependCHILD+LogReg	279	169	524	414	0.57	0.49
DependCHILD+SVM	210	116	577	483	0.57	0.41
DependCHILD+NB	34	8	685	659	0.51	0.10
BBTC+NB	171	121	572	522	0.53	0.35
BoW+LogReg	1	0	693	692	0.50	0
BoW+SVM	1	0	693	692	0.50	0
BoW+NB	693	692	1	0	0.50	0.66

2. **Elimination of underrepresented/noisy genres:** MAL recognize several niche-focused genres, such as “Gourmet” or “Boys Love”, who are very low representated in the dataset. We tackle that issue in an unwanted way, by merging them with highest represented genres. Is an unwanted measure because they are considered independent genres for a very good reason. In table 2 we explain those merges. If an anime lost all of its genres, we remove it from the dataset.

After that preprocessing, the amount of samples got reduced from $21,460$ to $10,168$. The genre proportion is depicted in Figure 4.

Golden label: We use a binary golden label as a baseline: animes with weighted score of 7 or higher are considered successful, while animes with score lower than 7 are labeled as unsuccessful. We arbitrarily decided that, guided by the fact that the dataset was splitted in an approximated proportion of 2:1 unsuccessful/successful by it (Figure 4). Moreover, the current value of C in march of 2023 is around 6.7, which is very near to 7.

Balancing the corpus: Given the preprocessed dataset with the associated binary golden labels, we implemented Algorithm 1 in order to balance it. Note that Algorithm 1 do not balance perfectly the genres, because sacrifies it in order to properly balance the golden labels, according to the value of μ. The new genres and golden label proportion after executing our implementation are depicted in Figure 5.

4.2 Experimental Setup

We splitted AniSyn7 in 80:20, having $5,542$ samples for the training data and $1,386$ samples for the test data. We used SVM, Gaussian NB and LogReg with the SciKitLearn’s default hyperparameters. We made the training with 5-fold cross validation.

BoW+SVM

	P	N
P	1	693
N	0	692

BoW+LogReg

	P	N
P	1	693
N	0	692

BoW+NB

	P	N
P	693	1
N	692	0

BBTC+SVM

	P	N
P	342	485
N	208	351

BBTC+LogReg

	P	N
P	336	502
N	191	357

BBTC+NB

	P	N
P	171	572
N	121	522

DCHILD+SVM

	P	N
P	210	577
N	116	483

DCHILD+LogReg

	P	N
P	279	524
N	169	414

DCHILD+NB

	P	N
P	34	659
N	8	685

Fig. 6. Confusion matrices.

5 Results

Results are depicted in Table 3. Confusion matrices can be found in Table 6. The vectorization with the worst performance was BoW, whose MLC's only predicted a single class. It improved when we added bigrams, trigrams and character trigrams in the BBTC vectorization, who suggests that the correlation between anime success and lexical features must be highly detailed in order to be significant. DependCHILD, on the other hand, had an average performance.

However, in NB, the model tended to predict only unsuccess, who suggests that the correlation between anime failure and syntactical features could be weak. This hypothesis must to be reinforced/invalidate through better syntactical vectorizations, in a similar way that we enhanced the BoW vectorization.

6 Conclusions and Further Work

In this paper, we presented the corpus AniSyn7 for the synopsis based anime success prediction task. From a MyAnimeList dataset gathered from Kaggle, we cleaned noisy data, defined a binary Golden Label and made a genre balancing with a view to a multitask setting, for further work.

We explored this corpus with three vectorizations, the first and the second (bag of words and bag of words + bigrams + trigrams + character trigrams) based in lexical features, and the third (Counting the childs on dependency trees) based in syntactical features. Through three MLC, we obtained several insights about the statistical correlation between syntactical and lexical features with success/unsuccess.

Further work will be to consider the multitask setting, and also to turn it into a multimodal project: early stages of development also includes character and element visual designs, so we plan to introduce image vectorizations to the task. We will also enhance our text-based baseline by adding more MLC's such as K-nearest neighbour and perceptron, as well as deep learning techniques such as recurrent neural networks and convolutional neural networks.

We also plan to upgrade the experimental setup by using different values of μ. Vectorizations will be also enhanced by adding word embeddings, attention mechanisms and transformers, in order to obtain a deeper understanding about the correlation between plot and success.

References

1. AlSulaim, S. M., Qamar, A. M.: Prediction of anime series' success using sentiment analysis and deep learning. In: International Conference of Women in Data Science at Taif University, pp. 1–6 (2021) doi: 10.1109/WiDSTaif52235.2021.9430244
2. Balouchzahi, F., Shashirekha, H. L., Sidorov, G.: Cosad-code-mixed sentiments analysis for dravidian languages. In: Central Europe Workshop Proceedings, vol. 3159, pp. 887–898 (2021)
3. Buda, J., Bolonyai, F.: An ensemble model using n-grams and statistical features to identify fake news spreaders on twitter notebook for pan at clef 2020. In: Conference and Labs of the Evaluation Forum (2020)
4. Cho, H., Schmalz, M. L., Keating, S. A., Lee, J. H.: Information needs for anime recommendation: Analyzing anime users' online forum queries. In: ACM/IEEE Joint Conference on Digital Libraries (2017) doi: 10.1109/jcdl.2017.7991602
5. Field, S.: Screenplay: The foundations of screenwriting. Delta Trade Paperbacks (2005)
6. Gemeda-Yigezu, M., Tonja, A. L., Kolesnikova, O., Shahiki Tash, M., Sidorov, G., Gelbukh, A.: Word level language identification in code-mixed Kannada-English texts using deep learning approach. In: Proceedings of the 19th International Conference on Natural Language Processing (ICON): Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, Association for Computational Linguistics, pp. 29–33 (2022)
7. Kim, Y. J., Cheong, Y. G., Lee, J. H.: Prediction of a movie's success from plot summaries using deep learning models. In: Proceedings of the Second Workshop on Storytelling, Association for Computational Linguistics, pp. 127–135 (2019) doi: 10.18653/v1/W19-3414
8. Maharjan, S., Kar, S., Montes-y-Gómez, M., González, F. A., Solorio, T.: Letting emotions flow: Success prediction by modeling the flow of emotions in books. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, vol. 2, pp. 259–265 (2018) doi: 10.18653/v1/n18-2042
9. Maharjan, S., Ovalle, J. E. A., Montes-y-Gómez, M., González, F. A., Solorio, T.: A multi-task approach to predict likability of books. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, vol. 1, pp. 1217–1227 (2017) doi: 10.18653/v1/e17-1114
10. Martín-del Campo-Rodríguez, C., Pérez-Alvarez, D. A., Maldonado-Sifuentes, C. E., Sidorov, G., Batyrshin, I., Gelbukh, A.: Authorship attribution through punctuation n-grams and averaged combination of SVM. Proceedings of the Conference and Labs of the Evaluation Forum, pp. 9–12 (2019)

11. Nuurshadieq, Wibowo, A. T.: Leveraging side information to anime recommender system using deep learning. In: 3rd International Seminar on Research of Information Technology and Intelligent Systems, pp. 62–67 (2020) doi: 10.1109/ISRITI51436.2020.9315363
12. Ojo, O. E., Gelbukh, A., Calvo, H., Feldman, A., Adebanji, O. O., Armenta-Segura, J.: Language identification at the word level in code-mixed texts using character sequence and word embedding. In: Proceedings of the 19th International Conference on Natural Language Processing (ICON): Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, Association for Computational Linguistics, pp. 1–6 (2022)
13. Olumide Ebenezer, O., Thang-Ta, H., Gelbukh, A., Calvo, H., Sidorov, G., Oluwayemisi-Adebanji, O., Armenta-Segura, J.: Automatic hate speech detection using deep neural networks and word embedding. Computacion y Sistemas, vol. 26, no. 2, pp. 1007–1013 (2022) doi: 10.13053/CyS-26-2-4107
14. Ortiz, G., Enguix, G. B., Gómez-Adorno, H., Ameer, I., Sidorov, G.: Job offers classifier using neural networks and oversampling methods (2022) doi: 10.48550/ARXIV.2207.06223
15. Pizarro, J.: Profiling bots and fake news spreaders at PAN'19 and PAN'20 : Bots and gender profiling 2019, profiling fake news spreaders on twitter 2020. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pp. 626–630 (2020) doi: 10.1109/DSAA49011.2020.00088
16. Reynaldi, Istiono, W.: Content-based filtering and web scraping in website for recommended anime. Asian Journal of Research in Computer Science, vol. 15, no. 2, pp. 32–42 (2023) doi: 10.9734/ajrcos/2023/v15i2318
17. Shahiki-Tash, M., Ahani, Z., Tonja, A. L., Gemeda, M., Hussain, N., Kolesnikova, O.: Word level language identification in code-mixed Kannada-English texts using traditional machine learning algorithms. In: Proceedings of the 19th International Conference on Natural Language Processing: Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, Association for Computational Linguistics, pp. 25–28 (2022)
18. Tonja, A. L., Yigezu, M. G., Kolesnikova, O., Tash, M. S., Sidorov, G., Gelbuk, A.: Transformer-based model for word level language identification in code-mixed kannada-english texts (2022)
19. Wang, P., Yan, Y., Si, Y., Zhu, G., Zhan, X., Wang, J., Pan, R.: Classification of proactive personality: Text mining based on weibo text and short-answer questions text. IEEE Access, vol. 8, pp. 97370–97382 (2020) doi: 10.1109/ACCESS.2020.2995905