50.045 Information Retrieval 1D Project: Anime Recommendation System

Aditya Vishwanath, Pang Bang Yong, Dong Jiajie

April 22, 2023

Abstract

This project details our attempts at building a text-based information retrieval engine for anime shows by processing natural language. The aim of the project is to find the most relevant anime shows, given a search query. In our work, we detailed n different approaches and examined their accuracy based on metric. We concluded that the chosen model outperformed the other model(s) just slightly, making it the most efficient model .

1 Introduction

Anime is defined as a style of Japanese film and television animation which is very popular amongst people of all ages today. Its is also a type of animation that has a huge plethora of shows and episodes for one to choose from. This also means that it makes it all the more harder for an anime enthusiast or new watchers to find a specific anime or to find new animes to watch after completing one. Text based information retrieval allows us to input a text search query to search through a data set and obtain the most relevant results to the query based on a similarity score. In our project, we harnessed this concept to allow one to search for an anime as well as ones related to it using text queries to solve the problem at hand.

2 Dataset

We obtained our dataset from Kaggle, called Anime Recommendation Database 2020. The database consists of **5 different CSV files:**

1. **anime.csv**
   - 35 columns
   - 17562 rows
2. **anime_with_synopsis.csv**
   - 5 columns
   - 16214 rows
3. **animelist.csv**
   - 5 columns
   - 109224747 rows
4. **rating_complete.csv**
   - 3 columns
   - 57633278 rows

5. **watching_status.csv**
    - 2 columns
    - 5 rows

We first used the anime.csv dataset to implement the basic Vector Space Model (VSM) Search Engine and then the anime_with_synopsis.csv to implement the Latent Semantic Analysis (LSA) and Best Match 25 Search Engines to take the synopsis into account.

3 Vector Space Model Search Engine

The Vector Space Model, or VSM, is an algebraic model for representing text documents as vectors of identifiers. It is often used in information filtering, information retrieval, indexing and relevancy rankings.It is one of the simpler types of models harnessed to build a search or recommendation engine. Having already implemented the Boolean Retrieval Engine in the first half of the course as an initial step, we moved to a slightly more advanced VSM engine as a building block in the process of us improving our project as well as reinforcing our learning. The model. This model can be overviewed with 3 main steps:

1. Represent words as vectors
2. Calculate tf-idf
3. Calculate Cosine Similarity between query and document vectors

$$\cos\theta = \frac{\mathbf{d_2} \cdot \mathbf{q}}{\|\mathbf{d_2}\| \, \|\mathbf{q}\|}$$

4. Rank documents based on relevance

4 Latent Semantic Analysis Search Engine

Latent Semantic Analysis is a technique in natural language processing (NLP), in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. This model assumes that words that are similar in meaning will occur in similar pieces of text.

1. Create matrix containing word counts per document
2. Apply Singular Value Decomposition (SVD) to reduce number of rows
3. Calculate Cosine Similarity between document vectors
4. Rank documents based on relevance

5 Best Match 25 Search Engine

Best Match 25 Model is a ranking model for evaluating the relevance between query terms and documents. Each word q_i in the query sentence will be calculated a relevance score with each document d, by weighting the relevance score for each word in the query statement for each document, we get the relevance score for the entire query statement for each document. Hence, there are three parts in the BM25 formula:

1. Correlation between each word q_i in the query and document d
2. Correlation between each word q_i with query
3. Rank documents based on relevance score

The normal BM25 formula is :

$$Score(Q, d) = \sum_i^n W_i R(q_i, d)$$

In which Q representing a query sentence, q_i representing a word from query, d representing a search document

W_i representing the word's weight

W_i is acutally the IDF of q_i:

$$IDF(q_i) = \log \frac{N - df_i + 0.5}{df_i + 0.5}$$

In which N representing the number of documents that will be searched, df_i representing the number of documents that contain word q_i. For a given q_i, the more documents contain q_i, the less important q_i will be, as the differentiation ability of q is very low. The IDF can therefore be used to characterise the relevance of q_i to a document.

Correlation between each word q_i in the query and document d

The design of BM25 is based on an important finding: the relationship between word frequency and relevance is non-linear, i.e. the relevance score of each word to a document does not exceed a specific threshold, and its impact does not increase linearly once the number of occurrences of the word reaches a threshold that would be related to the document itself. Therefore, when portraying word-to-document similarity, BM25 is designed in such a way that:

$$S(q_i, d) = \frac{(k_1+1)tf_{td}}{K+tf_{td}}$$

$$K = k_1(1 - b + b * \frac{L_d}{L_{ave}})$$

Where tf_{td} is the word frequency of word t in document d, L_d is length of document d and L_{ave} is the average length of all documents. The variable k is a positive parameter used to normalize the range of article word frequencies. Normally is set to 1. b is another adjustable parameter (0¡b¡1), which is used to determine the range of information content using document length: when b is 1, it is the full weighting of words using document length, when b is 0 it means that no document length is used. Normally is set to 0.75.

Correlation between each word q_i with query

When the query is very long, we also need to measure the weight between the words and the query. For short query, this item is not necessary. In our anime search engine, most of the use case are searching with short phrase, so we did not implement that feature in our BM25 model
By combining all the formula mention before, we get the formula for our BM25 model:

$$\mathrm{Score}(Q, d) = \sum_{i}^{n} \mathrm{IDF}(q_i) \cdot \frac{f_i \cdot (k_1 + 1)}{f_i + k_1 \cdot (1 - b + b \cdot \frac{dl}{avgdl})}$$

Both the query and documents in the corpus were pre-processed by:

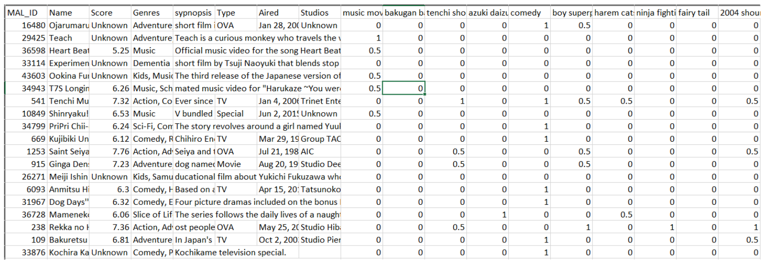
1. Removing stopwords
2. Converting letters to lowercase
3. Removing tags and special characters
4. Tokenizing

6 Evaluation

To compare the performance of the different models, various evaluation metrics were used on a self-crafted evaluation dataset.

6.1 Evaluation Dataset

As there are no readily available test dataset and the raw dataset is too large, we created our own evaluation dataset by taking a random sample of 50 data rows from the raw dataset as shown in Figure 1 below. We then manually crafted 10 different queries specifically for the evaluation dataset and labeled each data point's relevance to each query in a three-point scale: Relevant, Partially Relevant, Irrelevant.



MAL_ID	Name	Score	Genres	synopsis	Type	Aired	Studios	music mov	bakugan b	tenchi sho	azuki daizu	comedy	boy superp	harem cat	ninja fighti	fairy tail	2004 shou
16480	Ojarumaru	Unknown	Adventure	short film i	OVA	Jan 28, 20(	Unknown	0	0	0	0	1	0.5	0	0	0	0
29425	Teach	Unknown	Adventure	Teach is a curious monkey who travels the \				1	0	0	0	0	0	0	0	0	0
36598	Heart Beat	5.25	Music	Official music video for the song Heart Beat				0.5	0	0	0	0	0	0	0	0	0
33114	Experimen	Unknown	Dementia	short film by Tsuji Naoyuki that blends stop				0	0	0	0	0	0	0	0	0	0
43603	Ookina Fu	Unknown	Kids, Music	The third release of the Japanese version of				0.5	0	0	0	0	0	0	0	0	0
34943	T7S Longin	6.26	Music, Sch	mated music video for "Harukaze ~You wer				0.5	0	0	0	0	0	0	0	0	0
541	Tenchi Mu	7.32	Action, Co	Ever since	TV	Jan 4, 2006	Trinet Ente	0	0	1	0	1	0.5	0.5	0	0	0.5
10849	Shinryaku!	6.53	Music	V bundled	Special	Jun 2, 2015	Unknown	0.5	0	0	0	0	0	0	0	0	0
34799	PriPri Chii-	6.24	Sci-Fi, Com	The story revolves around a girl named Yuul				0	0	0	0	1	0	0	0	0	0
669	Kujibiki Un	6.12	Comedy, R	Chihiro En	TV	Mar 29, 19	Group TAC	0	0	0	0	1	0	0	0	0	0
1253	Saint Seiya	7.76	Action, Ad	Seiya and t	OVA	Jul 21, 198	AIC	0	0	0.5	0	0	0.5	0	0	0	0.5
915	Ginga Den	7.23	Adventure	dog named	Movie	Aug 20, 19	Studio Dee	0	0	0.5	0	0	0.5	0	0	0	0
26271	Meiji Ishin	Unknown	Kids, Samu	ducational film about Yukichi Fukuzawa wh				0	0	0	0	0	0	0	0	0	0
6093	Anmitsu Hi	6.3	Comedy, H	Based on a	TV	Apr 15, 20	Tatsunoko	0	0	0	0	1	0	0	0	0	0
31967	Dog Days"	6.32	Comedy, E	Four picture dramas included on the bonus				0	0	0	0	1	0	0	0	0	0
36728	Mameneko	6.06	Slice of Lif	The series follows the daily lives of a naught				0	0	0	1	0	0	0.5	0	0	0
238	Rekka no H	7.36	Action, Ad	ost people	OVA	May 25, 20	Studio Hib	0	0	0.5	0	0	1	0	1	0	1
109	Bakuretsu	6.81	Adventure	In Japan's	TV	Oct 2, 200	Studio Pier	0	0	0	0	1	0	0	0	0	0.5
33876	Kochira Ka	Unknown	Comedy, P	Kochikame television special.				0	0	0	0	1	0	0	0	0	0

Figure 1: Screenshot of Evaluation Dataset

6.2 Mean Average Precision (MAP)

The Mean Average Precision (MAP) is calculated by taking the mean of the average precision scores for each test query. We first had to calculate the Precision@K for each rank position of the relevant documents among the retrieved set, which we can then use to compute the average precision scores by taking the average over all Precision@K. This would be one of the better metric to evaluate the different models as the search engine can return multiple relevant documents/recommendations for each query.

6.3 Mean Normalised Discounted Cumulative Gain (NDCG)

As our evaluation dataset was labelled on a three-point scale, we could use also use the normalised discounted cumulative gain (NDCG) as an evaluation metric. The NDCG is useful for comparing queries that returns varying number of results of varying relevance. We first calculate the discounted cumulative gain (DCG) by summing the relevance scores of the retrieved documents while discounting the lower-ranked relevant documents by taking the base-2 logarithm of the relevance scores. The ideal DCG can then be calculated by sorting the retrieved results by relevance score and computing DCG again. The NDCG of a query is obtained by dividing DCG by the ideal DCG. We took the mean of the NDCG across all test queries to have a single point of comparison for the different models.

6.4 Mean Reciprocal Rank (MRR)

As some of the test queries only have 1 relevant data point, looking at the reciprocal rank could be a better metric as it only looks for the highest ranked relevant document. The MRR was obtained by taking the average reciprocal rank of all test queries for the evaluation dataset.

6.5 Latency

The time taken to process a query is also an important factor as there is an acceptable range of latency for users to consider the search engine usable. We used iPython's autotime feature to measure the time taken for each model to process the different test queries and took the average. Taking the reciprocal of the time taken, we can get the queries-per-second (qps) of the respective models.

6.6 Results

Table 1 below shows the results of our evaluation.

Model	MAP	NDCG	MRR	Latency (qps)
$\boldsymbol{VSM}$	0.553	0.559	0.600	35.7
$\boldsymbol{LSA}$	0.551	0.617	0.570	66.6
$\boldsymbol{BM25}$	0.8233	0.882	0.875	32.3

Table 1: Evaluation Results

As seen from the results, BM25 consistently outperforms the other models on all evaluation metrics, while the LSA model is the fastest.

7 Challenges

1. **Lack of Evaluation Dataset**
   As mentioned earlier, our dataset and database did not contain an evaluation dataset that came with anime.csv or anime_with_synopsis.csv to enable us to evaluate the performance of our search engines effectively.

2. **Limited Queries**
   Stemming from the first challenge, we had to build our own evaluation dataset which resulted in very limited queries that we could use as test queries since we had to make them ourselves and in a limited timeframe.

3. **Bias of Evaluation Dataset**
   Creating our ow evaluation dataset manually as well as test queries also had an additional challenge of bias. This is also because the evaluation dataset and queries were made by one team member which limits it to his inference and also possibly favours one engine more than the other. Hence resulting in a bias in the evaluation dataset.

4. **Long Runtimes**
   As with most data science, AI, CV and NLP projects and tasks, we also ran into instances where our implementations had painfully long runtimes which does hinder the rate of our project. With the right optimisation however, this can and was easily circumvented by our team as seen in the metrics of our engines.

5. **Limited Experience**
   Being new on the job is never easy and as familiar as we were with the concepts of various models and algorithms in information retrieval, we lacked the expertise to efficiently implement all of the models we had in mind for this project.

6. **Time and Schedules**
   Another non-technical challenge was our schedules and time constraints due to our modules apart from IR as well as the Goliath that is Capstone, which ate up a good portion of our time over the course of this term.

8 Future Works

1. **Adaptation to Japanese text**
   Anime is consumed all over the world but the primary and root consumer market is in Japan. This means that a lot of show titles would also be represented in Japanese text as well and users in Japan might even wish to search by Japanese text. Hence, a future work would be to make our engine adaptable to both English and Japanese queries.

2. **Neural Network based Search Engine**
   Due to the time and experience challenges mentioned above, we were not able to implement a search engine that harnessed Neural Networks to perform searches and recommendations despite having learnt about it conceptually. A future work would be to bring this to fruition and hence build an even smarter engine.

3. **Expand to animations worldwide**
   Apart from expanding our query capabilities to taking in Japanese text, another future work to make our solution even more extensive would be to go beyond anime and also allow our system cover animations from all over the world as well.

4. **Larger Evaluation Dataset**
   Going forward from the challenges we faced, most of which originated from the lack of an evaluation dataset and the small size of the one we created, a good thing to work on would be to make a larger evaluation dataset which could also be based of some user testing on how different people decided what kind of test queries would be associated with each title in the evaluation dataset, making the test queries more extensive as well and off setting the evaluation dataset bias mentioned in the challenges.

5. **Weight for various input data for BM25**
   A useful future work would be to assign weights to the different columns of the dataset we take in to search by. For example, possibly assigning higher weights to ”Name” or ”Genre” to make one of them more important.

6. **Experimenting with Ablation and accounting for it**

   The final future work would be to look into ablation. This means studying the effects of removing one column of data that we take in to search by and how it affects searches by other categories as well. This would also help us bette perform the previous work of assigning the optimal weights to each of these categories.

9 Conclusion

In this project we managed to build an effective Recommendation Engine for Anime shows using the Best Match 25 Model after implementing and evaluating ones using Latent Semantic Analysis and Vector Space Model as well. Although we did not experiment with as many models and algorithms as we would have liked to, we learnt a lot about the application of the concepts of these models, the importance and semantics of creating an evaluation dataset, how to choose the right evaluation metrics for an engine, as well as being able to learn from our shortcoming and innovate for the future.

10 Source Code

The code for our project can be found at: IR Project 2022