Astronomical Object Retrieval by Text or Image

1 Introduction

The amount astronomical observation data is vast and finding interesting objects to view can be a challenge. For amateur astronomers it can be difficult to find information about interesting objects to see without specific names or coordinates. Additionally sometimes an astronomer will notice something interesting and want to get more information about it and objects like it. While astronomical databases and star maps exist, they require specific details about what you are looking for in their queries, such as brightness, coordinates, or some specific scientific identifier. The goal of this project is to develop a system to retrieve position and appearance data on astronomical phenomenon given textual data or images.

To achieve this, we use a custom dataset built using structured databases and two machine learning models to both encode queries and score objects based on their image and relevant metadata. Using a subset of the dataset, the first machine learning model encodes images and their object metadata using a pretrained multimodal encoder (Clip). This allows us to handle image queries and text queries in a similar way. The second machine learning model is a learning to rank model (LambdaMart) that uses a set of known relevance scores to learn associations of queries and documents. This allows us to build a latent representation of objects and automatically map different types of queries to them.

In limited testing, we show that multi modal training is feasible for retrieving relevant documents. However, image based retrieval is significantly less discriminating than text and queries that include relevant metadata. Using a generic pretrained Clip models yield poor performance unless the query has some very specific and recognizable detail to add in classification. Additional training on a large amount of astronomical data may be required for better performance.

2 Data

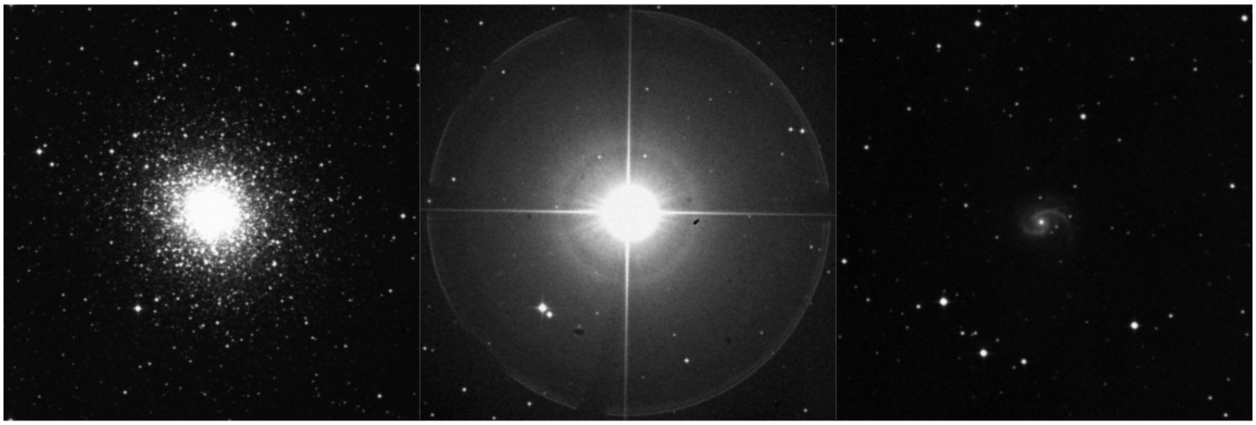
In order to generate an information rich latent space for celestial objects, we build a dataset of images, research articles, and object metadata. The core of the dataset comes from the SIMBAD Astronomical Database[1]. In order to capture only stars visible without high power equipment, we queried SIMBAD for all objects with magnitude 6.5 or less (eg. 6.5 or brighter) in either the red, green, blue, and broadband spectrum. This provides us with relevant object metadata, such as object names, right ascension, declination, object type, and magnitude. This query also provides a list of research articles associated with the object in the form of DOI entries. These entries are queried using the crossref API[2] and provides the corresponding document. For better breadth of information, only the title and abstract for up to 6 documents are recorded as object features rather than using the full text of a single document. This avoids some bias related to the subject of any given article as they often are not specifically on the object and may simply include it in a survey. Objects without any retrievable articles are included in the dataset, but have limited reachability for the text based

[1] Wenger, Marc, et al. "The SIMBAD astronomical database-The CDS reference database for astronomical objects." Astronomy and Astrophysics Supplement Series 143.1 (2000): 9-22.

[2] https://www.crossref.org/documentation/retrieve-metadata/rest-api/

portion of our system and serve mainly has training data.

The dataset also contains images for each object. Images are generated using the hips2fits service[3] based on object ID and position. Images are stored in 500x500 greyscale pngs.



The list of features are: an object number, object names (ex M31 is Andromeda), right ascension, declination, magnitude (taken as brightest in each available color channel), object type, article title, article abstract, and images. The composition of the data is as follows:

# Objects	# Features	# Object Types	Documents	Images
12972	8	80	90%	100%

Test relevance scores are required for both image queries and text based queries. For image based queries, a subset of the dataset images can be used as queries. The relevance score can be

[3] https://alasky.u-strasbg.fr/hips-image-services/hips2fits

described as an inverse loss function of the type of object t (one hot vector whose entries correspond to categories star, galaxy, nebula, etc.), the brightness of the object m, and the separation of the objects in arcseconds s: $L(t, m, s) = (t^T t*) + 0.2|m - m*| + 0.05 * s$. This is then normalized to a score 1-5.

Document relevance scores for text based queries are done more subjectively. Thirty sample text queries are generated and the corpus is scored using BM25 on the object articles to get the top fifty documents for each query. For queries that include an object name, high relevance scores are awarded to objects that are in the query or are subset of those objects (a star in orions belt is nearly as relevant as orion's belt itself in most cases). Some example queries and relevance data are shown as follows:

Queries:

"what is the closest star to the north star",
"show galaxies visible from the northern hemisphere",
"visible galaxies",
"the brightest star",
"stars closest to the andromeda galaxy",
"easiest to see galaxy",
"nebula",

Scores:

query,doc,rel
where is the andromeda galaxy,6193,5
#Note object 6193 corresponds to the Andromeda Galaxy
where is the andromeda galaxy,281,4
#Note object 281 corresponds to a star in the andromeda constellation
where is the andromeda galaxy,6213,3
where is the andromeda galaxy,4532,1
where is the andromeda galaxy,341,3
where is the andromeda galaxy,515,1
where is the andromeda galaxy,3357,2
where is the andromeda galaxy,512,4
where is the andromeda galaxy,3663,2
where is the andromeda galaxy,459,1

3 Related Work

Typically astronomical information is kept in structured databases. For example, SIMBAD and SDSS[4] are SQL databases and can be queried by specific identifiers of object in question or by ranges of relevant statistics. This makes it very easy to find information if you know what you are looking for, but not very helpful in general queries. When an object is discovered it is given a unique identifier. Only objects that are part of a further study or are large and unique are given names. However, names are a typically user will query for an object. Would a typical person

[4] Almeida, Andrés, et al. "The Eighteenth Data Release of the Sloan Digital Sky Surveys: Targeting and First Spectra from SDSS-V." The Astrophysical Journal Supplement Series 267.2 (2023): 44.

want information about "Messier 31" or "The Andromeda Galaxy", which are one and the same. Some proprietary starmap software, such as Stellarium[5], will use a custom database where popular objects will be given more information along with real images, but most objects will have minimal information other than coordinates and its scientific identifier. These star maps also require the user to know where or what they are looking for.

For unstructured data, there is very little similar work. The closest related work was present in "An Information Retrieval and Recommendation System for Astronomical Observatories"[6]. The authors of this paper aim to build natural language process algorithm using machine learning to search observatory logs to present objects of interest related to a query. This approach is similar to this project in that it tokenizes text queries to search for astronomical data, but is strictly related to text based retrieval. There is significantly more research in the classification objects using machine learning[7][8][9]. However, the intended purpose of these models is to automatically classify unknown images into set object types. In "Machine learning classification of SDSS transient survey images"[10], the authors successfully classify images from the SDSS database into "real supernova" and "artifact" classes using a combination of principal component analysis and trained support vector machines. This project doesn't aim to classify images, but give relevance scores based on a given query.

4 Methodology

A preliminary method for determining feasibility is to rank object from text queries using a Bm25 ranker. We first tokenize the abstracts of each object and append their names. We then use the rank-bm25 OkapiBM25Model library[11] to build an index using these tokenized texts and score query-object pairs. This method was only used in scoring text based queries, but provides a useful feature in such cases.

To train and test our full system, we use a collection of document features generated per query, including the BM25 score. The features we use include: the BM25 score for the abstracts, the BM25 score for the names, the cosign similarity of the query to image CLIP encoding, the similarity of the query to the abstract title CLIP encoding and the object magnitude. To encode the query and object images, we use the pretrained openai/clip-vit-base-patch32 CLIPmodel, CLIPImageProcessor and CLIPTokenizer from the transformers library[12]. This allows use to convert objects and queries into a latent vector space that can be directly compared. Object magnitude is included as a feature because we determined that higher object brightness typically corresponds to more relevant documents when generating the training relevance score document described in section 2. These features are then passed to a LGBRanker instance trained on the training relevance score document for scoring.

[5] https://stellarium.org/

[6] Mukund, Nikhil, et al. "An information retrieval and recommendation system for astronomical observatories." The Astrophysical Journal Supplement Series 235.1 (2018): 22.

[7] Bertin, E. "Classification of astronomical images with a neural network." Astrophysics and Space Science 217 (1994): 49-51.

[8] Odewahn, S. C. "Automated classification of astronomical images." Publications of the Astronomical Society of the Pacific 107.714 (1995): 770.

[9] Brunner, Robert J., et al. "Massive datasets in astronomy." Handbook of massive data sets (2002): 931-979.

[10] Du Buisson, L., et al. "Machine learning classification of SDSS transient survey images." Monthly Notices of the Royal Astronomical Society 454.2 (2015): 2026-2038.

[11] https://pypi.org/project/rank-bm25/

[12] https://huggingface.co/openai/clip-vit-base-patch32

For image based queries, we only use the query CLIP encoding and compare it to the encoding for each object image and each object's abstract titles. The reason we do not encode entire articles into the latent space is the available pretrained model is only designed for short queries and image captions and thus has a 77 character limit. As the articles contain significantly more characters than this, they were excluded. Additionally, this encoding and evaluation of the CLIP model is very computationally expensive, so switching to a larger model wasn't feasible due to extremely long query times.

5 Evaluation and Results

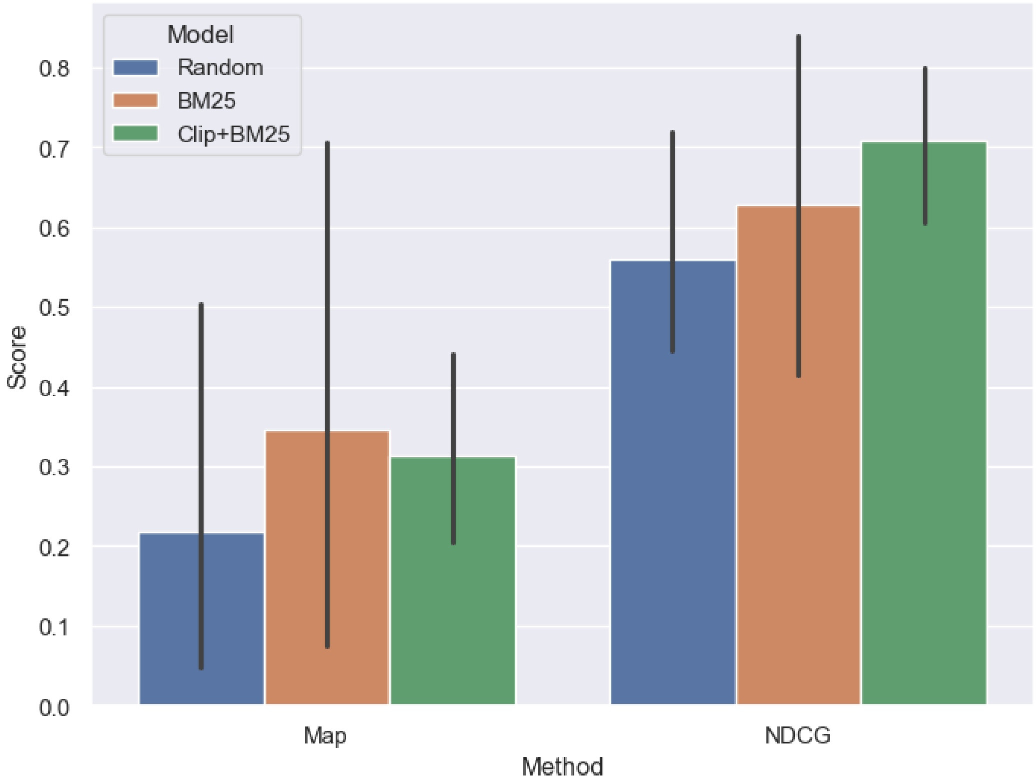
To evaluate the performance of the system, we use two simple baselines to compare against. The first baseline is to randomly select objects for a given query. This method is the most crude, but gives a performance floor for any query type. Because random scoring is independent of input, we can extrapolate its performance from either text based queries or image based queries and expect identical expected performance. A more competitive baseline was to score objects using the BM25 ranker described in section 4. This method provides better performance and a more realistic representation of how a user would search celestial object research articles. However, this method is for text only.

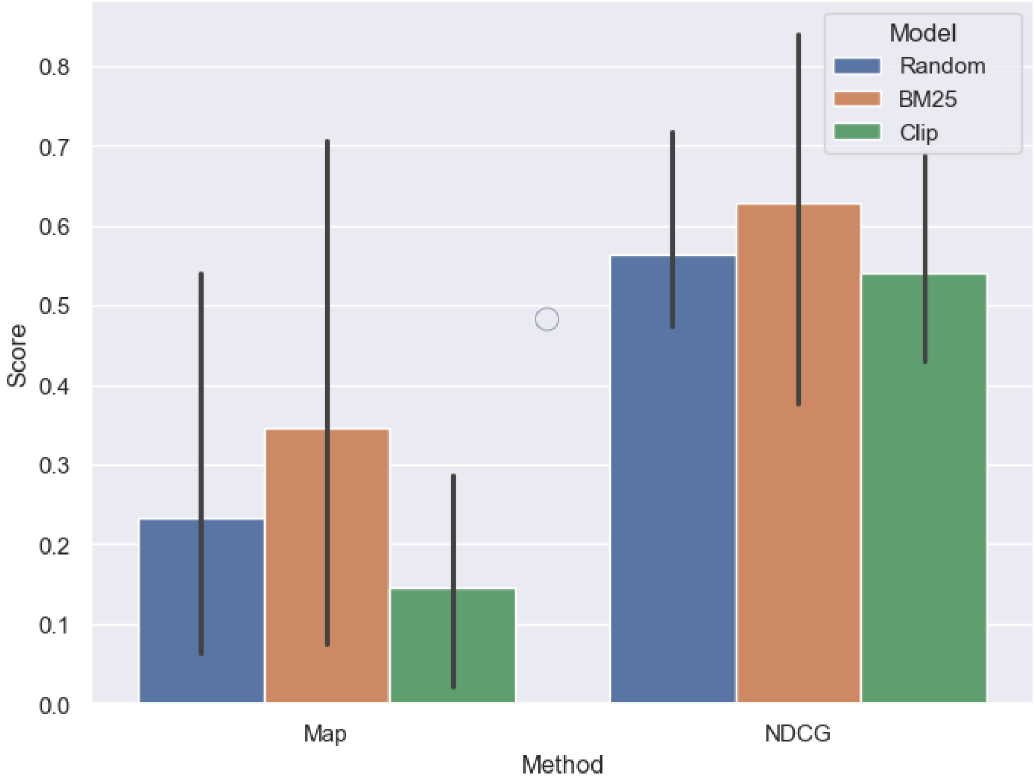
The relevance scores described in section 2 are done on documents gathered using the BM25 scorer described in section 4. This means that results of a BM25 recommendation system returns results for those queries in the order that they appear in the test relevance score document. These results are then randomly shuffled 50 times for each query and recorded as potential results from a random ranker. This data is used for evaluating the baseline

To evaluate our trained LGBRanker, we evaluate scores only for documents available in the relevance scores document. This effectively demonstrates the performance of a system that initially ranks all documents and then reranks the top 50 documents. We do this as we found that the CLIP based scorer and ranking model were slow to evaluate on the entire corpus in a reasonable amount of time. The documents are then sorted by their score and given relevance ratings equal to what is found in the relevance scores document.

We then use the MAP@10 score for these documents with their given relevance scores converted into binary values, with scores greater than 3 as relevant and scores less than 3 as not relevant. We also use the NDCG@10 metric to determine whether the system provides reasonable ordering to the returned objects. Testing yields a MAP and NDCG score of approximately 0.34 and 0.63 respectively for the BM25 scorer, 0.21 and 0.55 for the random scorer, and 0.31 and 0.71 for the learned model.

We also include the performance of a CLIP only scorer to demonstrate the value of multimodal features when scoring. This scorer performs worse than even random selection with MAP and NDCG score of just 0.15 and 0.54 respectively.


Model
Random
BM25
Clip+BM25
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Score
Map
NDCG
Method


Model
Random
BM25
Clip
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Score
Map
NDCG
Method

6 Discussion

The baseline scores have a very large variance. One cause for this the different characteristics of the measured queries. When queries include the name of an object, then the BM25 scorer performs very well which in turn improves the performance of the learned ranker. When the query only includes visual descriptors (like "a bright object") the two differences occur: the BM25 ranker performs significantly worse as the frequency of these terms are not dependent on the object itself and the relevance of documents becomes less clear. For example, if the query is "a bright star", most objects in the dataset fall under this category and therefore most rankings (even random ones) will have a high MAP and NDCG score.

Adding the additional features helps to reduce the variance in these metrics by giving some additional context that might be missed in the text descriptions alone. We found the CLIP model was very good at scoring based on visual descriptions and very bad in all other cases. Again using the "a bright star" query as an example, CLIP is able to identify a large white spot in an object's image and yield a high similarity with the query (which in this case is reinforced by the magnitude feature). CLIP is also good at identifying dramatically different object types, such as when an image as galaxy in it versus a star as they look noticeably different. These are cases when BM25 struggles, so adding in the CLIP model helps to improve performance in such cases. However, when queries don't have visual identifiers, the CLIP model is not helpful in scoring and effectively adds noise to the scoring which reduces performance. Combined, the scorer achieves similar average performance to a BM25 scorer, but with much less variance.

For image based queries, the system performs poorly due to over-reliance on the general purpose CLIP model. The CLIP model used was trained for a wide variety of images and classes. Using it in the limited setting of astronomy images results in it struggling to separate objects of similar type. For example, it can separate stars from galaxies, but not the North Star from Sirrius (which would be even hard for a human given only an image and no further context). Without the additional context of image captioning, the system can't distinguish most images in the dataset.

7 Conclusion

We show how the addition of CLIP embeddings and document metadata can improve the consistency of object retrieval. However, image based retrieval is still a significant challenge. Using a generic pretrained Clip models yield poor performance unless the query has some very specific and recognizable detail to add in classification. In general astronomical object information retrieval, more features helps a learned ranker to be separate objects that appear very similar without additional context.

8 Other Things We Tried

To address the poor performance of the CLIP model, we investigated retraining the model using our dataset. This stems from the idea of using transfer learning, where model trained on a large dataset can be fine tuned to a new application with a relatively small dataset. This works because the large model has theoretically learned useful features in the large dataset and only needs to learn to convert them into the new application space. Using the first 10,000 objects and abstract titles as a training dataset, we attempted to further train the CLIP model for three epochs using the torch library. Instead of improving performance, the model began to act more randomly. This is potentially caused by the image and text encoders no longer being aligned in the models vector space. It could also be caused because we allowed all parameters to update (slowly) instead of freezing certain important layers.

9 What We Would Have Done Differently

The biggest change we would have made to our approach was to gather data on more objects and more details for each object. We intended to search only visible objects, but including more objects would allow us to train our learning models more effectively, even if we didn't include those objects in the searchable corpus. One reason for not including more objects is time constraints. It takes approximately 10 seconds per object to query its metadata and retrieve images and articles. For the nearly 13,000 objects in the current dataset, this took about 36 hours. For a complete dataset, we would want on the order of 100k or 1M objects which would take a month to a year given our implementation, which wasn't feasible. We also would have recorded more object metadata like object dimensions, distance from earth, survey details, and others that weren't immediately obvious were relevant, but could improve the performance of the learned ranker for certain queries.

10 Teamwork Statement

N/A