Real Estate Recommendation using Sentence Embeddings and Clustering*

M Srinivasan
IMT2021058
m.srinivasan@iiitb.ac.in

Siddharth Kothari
IMT2021019
siddharth.kothari@iiitb.ac.in

Munagala Kalyan Ram
IMT2021023
kalyanram.munagala@iiitb.ac.in

Sankalp Kothari
IMT2021028
sankalp.kothari@iiitb.ac.in

***Abstract*—This work aims to create a real estate recommendation system to recommend properties from 15 cities in USA. The recommendation is done based on features of the house, as well as features of the neighbourhood of the house, and aims to exploit images as well as descriptions provided by agents to give comprehensive and coherent recommendations to users, which align with most of their interests.**

***Index Terms*—sentence embedding, image captioning, real estate, clustering, recommender systems**

I. INTRODUCTION

Housing is one of the most basic and important needs for any individual or family, and constitutes one of the three basic needs for any human - food, clothing and shelter. There has been a recent surge in the use of machine learning based algorithms in this domain.

In the current climate of the Internet, there exist many sites which provide the user with real estate listings, based on a number of factors set by the user. We aim to model one approach in this work to recommend real estate properties which match most of the specifications provided by the user about the property which he desires.

There can be users with a whole spectrum of interests and necessities while buying/renting a house. Some users prefer the specifications of the house itself, which can include number of bedrooms, bathrooms, and the square footage of the house, while for others, the connectivity of the house to basic amenities such as schools, transit systems (buses, train stations etc.) may be of paramount importance, and they may be willing to negotiate on other parameters.

Another feature of such a system is that it should be able to take into account important features of the house which may be listed in the description and images provided for the house. These can include the presence of swimming pools, wooden floors, and high ceilings, which are not directly present in systems as parameters on which the user can filter the houses, but are rather present in the descriptions provided to the users. In this case, the user will have to read many descriptions to be able to filter out houses which cover most of these desired features.

Our system aims to address all these needs of the user, and then build a GUI based application to show the listings for each user. We use unsupervised learning methods to first assign a cluster to the user, and then within the cluster, we use sentence embeddings generated by a language transformer encoder, and compare similarities to recommend the most similar items to the user.

II. PREVIOUS WORKS

One recent work aims to use transformers combined with self supervision learning to be able to predict the price of a property. [1] Similar to our dataset, this dataset also has images of the interiors and exteriors of the houses, along with information on the bedrooms, bathrooms and square footage of the house. Our work draws inspiration from this one, but with a different end goal.

One work aims to survey various algorithms used in real estate recommendation, ranging from simple models such as collaborative filtering, content based filtering, to approaches based on more complicated methodologies such as Reinforcement Learning, and Multi Criteria Decision Making. [2]

One of the biggest problems faced in this domain, which we also encountered in our work, is the lack of sufficient user-item interactions in this domain. There are not many user reviews for any listing, and hence we are forced to rely on Unsupervised methodolgies such as clustering.

There have, however, been recent works which aim to address this problem. A good example of the same would be the approach used by Zhang et al. [3] They also make use of content based filtering, along with boosting tree models, to be able to tackle this problem.

III. DATASET ACQUISITION

A. Obtaining Listing Info

We obtained the data of listings using the Zillow website for the following 15 cities - New York City, Chicago, Los Angeles (LA), San Jose, Boston, Washington, D.C., Miami, Seattle, Houston, Dallas, Las Vegas, Atlanta, Philadelphia, Phoenix, Detroit.

We ended up with a total of 6936 listings of houses across the 15 cities, which serves as our initial dataset. Each row has a number of attributes such as no. of bedrooms, bathrooms, price, estimated rent, house area, latitude, longitude, home insights (i.e. what the house contains, such as garage, playground etc), house images and address info, and descriptions provided by the seller/real estate agent.

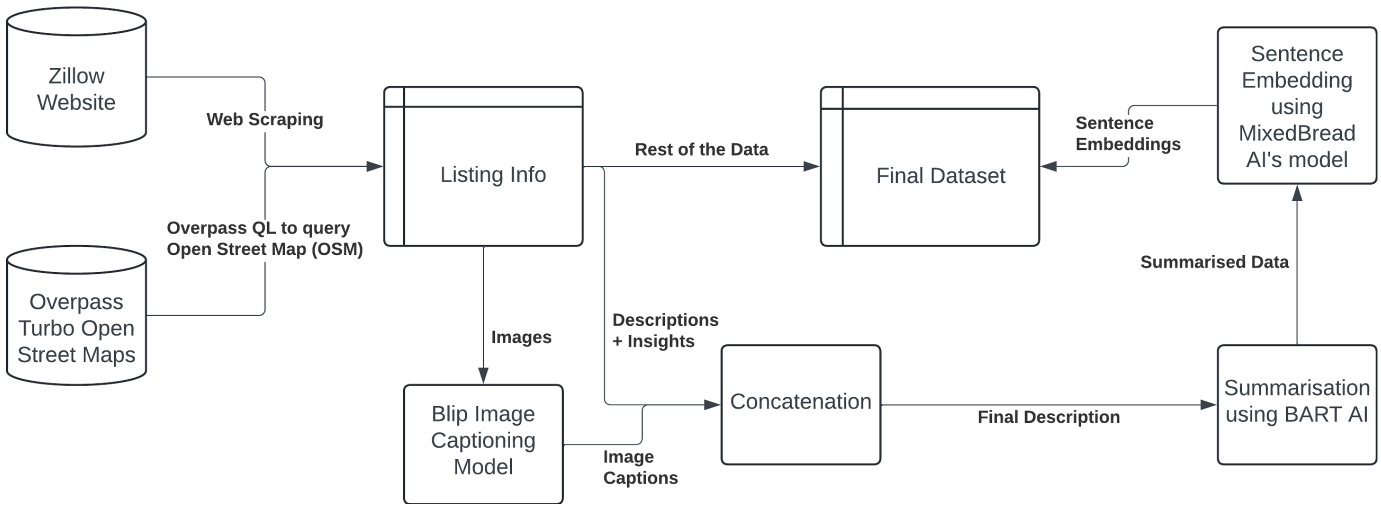

Fig. 1. Dataset Generation Workflow

B. Supplementing Data with Neighbourhood Info

Depending on the user, more than the house itself, the neighbourhood of the house may matter more. The user may value the connectivity of the house, with schools, leisure areas (parks, malls), shops and transit centers (bus stops, subway stations etc.) more than the house itself.

To take care of this, we queried the Overpass Turbo website, to get the info of schools, parks, malls (parks and malls are clubbed under "Leisure areas"), shops and transit centres in each of the fifteen cities.

Then for each data point, we annotate it with the number of leisure areas, schools, shops (departmental stores etc) within 5 km and transit centres within 2 km. This can serve as a measure of how well connected the listing is in each of these areas, and can be compared with the user's preferences for these values.

C. Image Captioning and Description Concatenation

From the zillow website, we got a list of images for each of the listing. We saw that the description given on the zillow website wasn't accurate for describing the house. More often than not, it included more info about the "Open Houses" for the listings, rather than the houses itself.

Hence we passed the images of each of the listing into a image captioning model to get more intricate details about the house. The info obtained from the description, image captions, and insights of the house are then concatenated to get a final long description of the house. We used Blip Image Captioning model by Salesforce. [4]

D. Text Summarising

The descriptions thus obtained ranged from 1000 characters to even as large as 14000 characters. We were unable to work on this using any model.

Hence we first had to run a summariser model, which would take 1024 characters at a time from the description, and then create a summary for those characters, and then do the same for each chunk of 1024 characters in the description.

Finally we just combined the summaries generated by each block, and get a final summary. This is done for each of the data points. We used Meta's BART AI model for the summarisation. [5]

E. Sentence Embeddings

We find similarity between the embeddings of the sentences in our recommendation algorithm (the details of which are provided later). For this, we generate and store the sentence embeddings of the description summaries which we generated in the previous step.

While recommending, we ask the user small descriptions of what he would want in the house, and the embedding of this description would be compared with the stored embeddings that we have computed in this step, and houses which are most similar in embeddings are recommended.

The embeddings are generated using MixedBreadAI's mxbai-embed-large-v1 model. [6]

F. Dataset Generation Workflow

A flowchart representing the entire data generation process is shown in Figure 1.

IV. Recommendation and Application Overview

A. Recommendation Algorithm

From the final dataset, we exclude the description, then with the remaining data, we split it based on the state.

For each state, we apply k-means++ clustering and determine the optimum k using the elbow method. The reason for this is simple. The clusters may represent different sets of houses across different states, and it wouldn't be desirable for a user to be recommended houses in different states from the one which he is searching properties in.

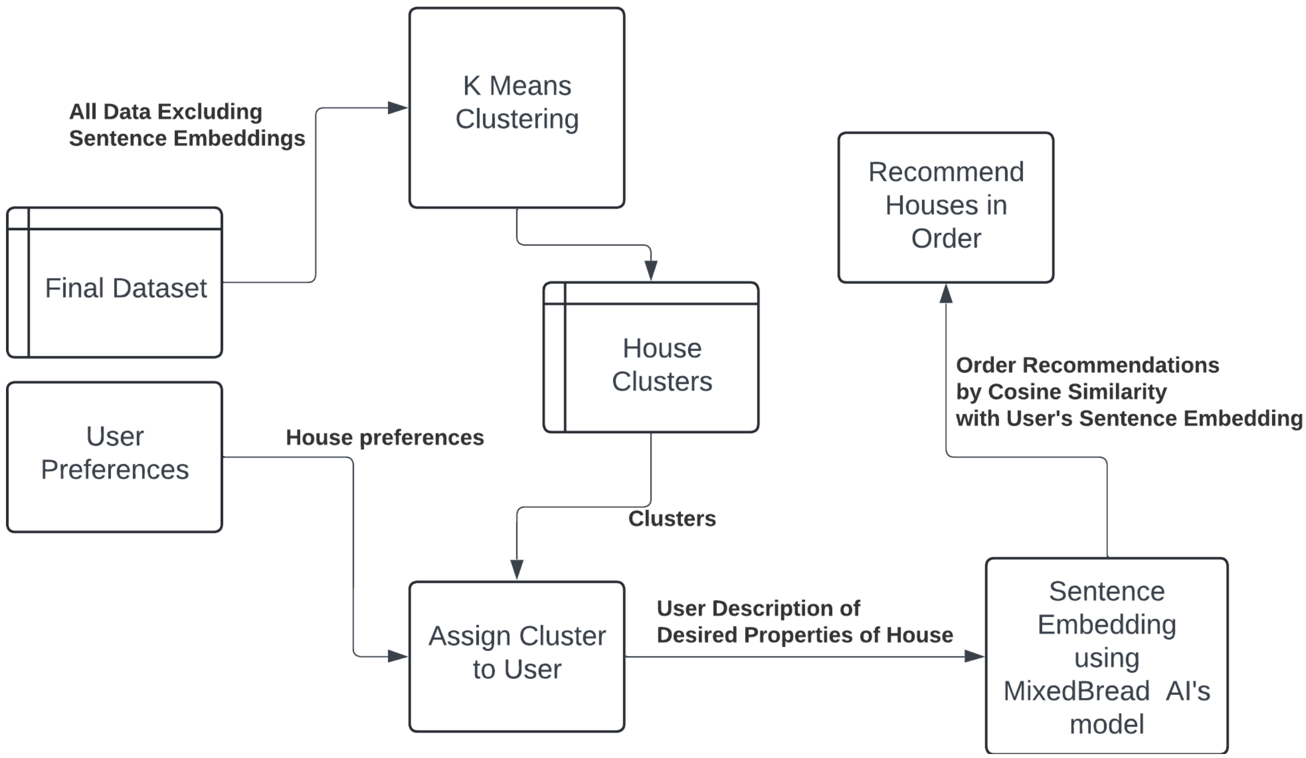

Fig. 2. Algorithm Workflow

The sentence embeddings for each description has been stored in the previous step separately. We then perform clustering for the listings for each state separately.

Once we have the clusters, we take the user input including description and state. We then find the appropriate cluster for the user. We then find the sentence embedding for the user's description and apply cosine similarity with the sentence embeddings of houses present in the cluster.

The final recommendation is done by displaying the houses in the user's cluster in decreasing order of the similarity score.

B. Algorithm Flowchart

A flowchart representing the algorithm is shown in Fig. 2.

C. Application Overview

We have implemented an application to allow users to interact with our recommendation model. The frontend of the application is implemented using "streamlit" where we created 2 pages - a form page which takes the user input, and the results page where the recommendations are displayed. The user can view the house details along with the images.

The application asks the user for the information on what all he expects from the house and ask him to give him a small description of the house he expects. Our website, takes into account the details mentioned by the user and then provides the users with the list of recommendations.

The user is first assigned an initial cluster according to his preferences. The houses in that cluster are recommended to the user. The order of recommendations is decided by the similarity score between the description given by the user and the description of the houses.

We implemented a small flask backend to host the sentence embedding model. The backend algorithm uses the transformer model to get the sentence embeddings of the user's description and performs the cosine similarity. The output is returned to the user, in order of decreasing similarity with the user's description.

References

[1] Yazdani, Mahdieh & Raissi, Maziar. (2023). Real Estate Property Valuation using Self-Supervised Vision Transformers. 10.48550/arXiv.2302.00117.

[2] Gharahighehi, Alireza & Pliakos, Konstantinos & Vens, Celine. (2021). Recommender Systems in the Real Estate Market—A Survey. Applied Sciences. 11. 7502. 10.3390/app11167502.

[3] Q. Zhang, D. Zhang, J. Lu, G. Zhang, W. Qu and M. Cohen, "A Recommender System for Cold-start Items: A Case Study in the Real Estate Industry," 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Dalian, China, 2019, pp. 1185-1192, doi: 10.1109/ISKE47853.2019.9170411.

[4] Li, Junnan, Dongxu Li, Caiming Xiong, and Steven Hoi. "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation." In International conference on machine learning, pp. 12888-12900. PMLR, 2022.

[5] Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461 (2019).

[6] Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. Open source strikes bread - new fluffy embeddings model.