FArMARe: a Furniture-Aware Multi-task methodology for Recommending Apartments based on the user interests

Ali Abdari
University of Udine
Via delle Scienze, 206, Udine, Italy
University of Naples Federico II
C.so Umberto I, 40, Napoli, Italy
abdari.ali@spes.uniud.it

Alex Falcon
University of Udine
Via delle Scienze, 206, Udine, Italy
falcon.alex@spes.uniud.it

Giuseppe Serra
University of Udine
Via delle Scienze, 206, Udine, Italy
giuseppe.serra@uniud.it

Abstract

Nowadays, many people frequently have to search for new accommodation options. Searching for a suitable apartment is a time-consuming process, especially because visiting them is often mandatory to assess the truthfulness of the advertisements found on the Web. While this process could be alleviated by visiting the apartments in the metaverse, the Web-based recommendation platforms are not suitable for the task. To address this shortcoming, in this paper, we define a new problem called text-to-apartment recommendation, which requires ranking the apartments based on their relevance to a textual query expressing the user's interests. To tackle this problem, we introduce FArMARe, a multi-task approach that supports cross-modal contrastive training with a furniture-aware objective. Since public datasets related to indoor scenes do not contain detailed descriptions of the furniture, we collect and annotate a dataset comprising more than 6000 apartments. A thorough experimentation with three different methods and two raw feature extraction procedures reveals the effectiveness of FArMARe in dealing with the problem at hand.

Keywords— Cross-Modal Understanding, Tag-aware Visual Representations, Metaverse Applications, Apartment Recommendation, Contrastive Learning

1. Introduction

Nowadays, it is common to move across different cities or nations every few years when looking for commercial or educational opportunities. This situation leads to an even more difficult problem, which is finding a new home. To do so, one has to read hundreds or thousands of different advertisements which often highlight the positive aspects of the house, apartment, flat, etc[1], while neglecting the negative ones (e.g., unappealing due to mismatched styles, gloomy due to unsightly furniture and colors, etc). Moreover, even when selecting a subset of the apartments and visiting only

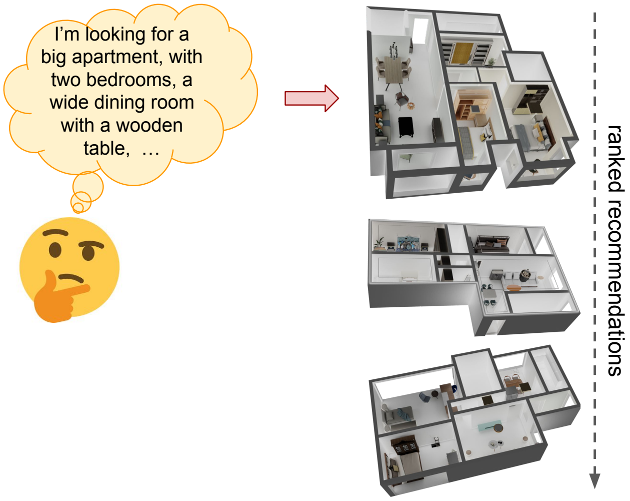

I'm looking for a big apartment, with two bedrooms, a wide dining room with a wooden table, ...
ranked recommendations

Figure 1: The proposed problem: text-to-apartment recommendation requires ranking all the apartments to recommend those which are most suitable based on the user interests.

[1] In the rest of the paper, we will use apartments to cover these slightly different situations.

those, the process is incredibly time-consuming and stressful, while also becoming more costly and environmentally unfriendly due to the need to physically move to the location. Therefore, visiting digital twins of the apartments in the metaverse would be a more favorable and ideal solution. However, we argue that the dedicated Web applications which are commonly used to ease the search process across all the advertisements, do not represent a good fit. In fact, it is common for these applications to identify a set of properties (e.g., availability, size, price, number of rooms, number of bathrooms, etc) to obtain a filtered list of viable candidates. Yet, these approaches do not allow for a comprehensive analysis of the apartment, as they enable specifying some *properties* but many *unstated user requirements* are left out. Therefore, to truly support the user in the search for an optimal apartment through metaverse exploration, there is a need for an automatic solution which both analyzes the contents of the metaverse scenario and the users' interests to find fewer recommendations that the user can comfortably visit.

The proposed problem, which we name as "**text-to-apartment recommendation**", requires to rank a set of apartments based on their relevance to a given user query, as shown in Figure 1. Interestingly, neither this problem nor its more general version related to retrieving 3D scenes based on textual queries, have been studied in the literature. The closest problem which was previously addressed consists in ranking 3d scenes based on 2d images used as queries [2, 3]. However, such a methodology leads to possible limitations. First, the user needs to have a visualization of the desired apartment, which may be difficult to have, while also limiting the flexibility of the approach (e.g., if the user sketches a floorplan, then it may be difficult to retrieve those apartments which have the same furniture and structural details, yet following a different layout). Second, it also limits the usability from the users' point of view, since describing the desired features in an apartment may be easier than sketching it, while also allowing for more flexibility both with respect to floorplan layout (e.g., by not mentioning the position of the furniture) and to the details which are left out in the query (e.g., if the user does not mention a need for a microwave oven, the retrieved apartments could include both those equipping it and those who do not). The proposed problem, which will be formally described in Section 2 along its main challenges, aims at allowing the users to describe their interests with free-form text, since this may represent a more comfortable solution, while also avoiding the limitations related to the more traditional Web-based approaches.

How to tackle this problem? We consider two main factors to support our design choices for a text-to-apartment model. First, the metaverse apartments could be seen as 3d scenes; however, the advertisements for them commonly contain several photos to capture the environment and the furniture available. Second, in recent years there have been impressive advancements in vision and language understanding [33], enabling several multimodal applications and allowing for cross-modal interactions. Therefore, we model our metaverse apartments as a set of images, annotate each apartment with very detailed descriptions, and use a CLIP-based approach to model the cross-modal relations and enable ranking. In particular, we implement several solutions based on deep learning, both inspired by recent literature and from past literature on related topics. The final methodology we propose, which we called FArMARe (Furniture-AwaRe Multi-task methodology for Apartment REcommendation), consists of a multi-task learning approach based on two sub-tasks. First, a contrastive learning objective requires the model to learn cross-modal relations helpful for ranking. Second, a furniture-based classification objective requires the model to more precisely identify the furniture displayed in the images. To experimentally validate our methodology, we collect a benchmark dataset consisting of around 6000 apartments, each of which is annotated by detailed descriptions of its furniture. This is because, as described below, the publicly available datasets do not offer both photo-like indoor pictures and detailed textual descriptions, which are required in our setting.

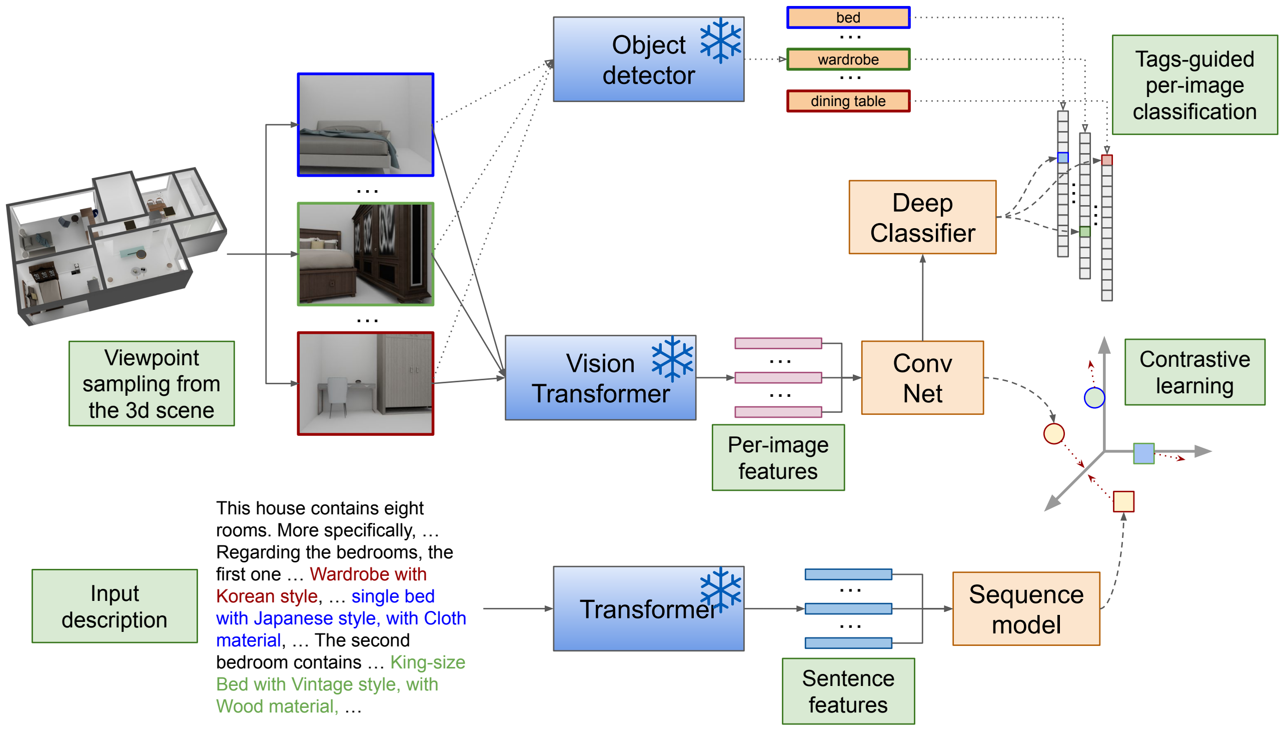
In summary, the following are the main contributions of this paper:

- We introduce the problem of text-to-apartment recommendation, and collect and annotate a benchmark dataset of more than 6000 apartments.
- We design a multi-task learning methodology, called FArMARe, to tackle the problem by introducing a furniture-aware training objective.
- We compare the effectiveness of our method with several baselines, inspired by previous research works on similar topics, and show its effectiveness, achieving an improvement of +1.1% R@1 and +2.0% R@5.

In Section 2 we describe the proposed text-to-apartment recommendation problem and its challenges. Then, Section 3 introduces the scientific literature related to our problem. The proposed methodology is described in Section 4, whereas a thorough experimental setting is followed in Section 5, and several limitations and future directions are highlighted in Section 6. Finally, Section 7 concludes the manuscript.

2. Text-to-apartment recommendation: definition and challenges

The proposed problem, text-to-apartment recommendation, requires understanding the contents of the 3d scenario



This house contains eight rooms. More specifically, … Regarding the bedrooms, the first one … Wardrobe with Korean style, … single bed with Japanese style, with Cloth material, … The second bedroom contains … King-size Bed with Vintage style, with Wood material, …

Figure 2: Overview of the proposed methodology. Details in Section 4.

representing the apartment, analyzing the user interests, and then connecting them. If the latter are formalized through *free-form* natural language, then the problem can be formally defined as follows. Given a set $\mathcal{A}$ of apartments, $\mathcal{A} = \{a_1, \dots, a_N\}$, each annotated by a description d_i, the objective is to learn two functions, f and g, such that the apartment a_i is the first retrieved if the query is d_i, i.e., $\forall j \neq i, sim(f(a_i), g(d_i)) \geq sim(f(a_i), g(d_j))$, where sim is a similarity metric. In our setting, inspired by the typical structure of the advertisements, each apartment a_i is depicted with a finite set of images.

The proposed problem could be considered similar to other instances of multimedia retrieval, such as text-to-image retrieval [33, 35]. However, two main intertwined challenges remain. First, there are many details of the apartment which need to be considered, both structural, e.g., how many rooms it has, the presence of separate living and dining rooms, etc, and furniture-related, such as all the modern appliances in the kitchen (fridge, microwave oven, etc). All these details can be inferred from visual analysis, and assessing their relevance to the user interests is very challenging both due to their amount and their different granularity. A second challenge is given by the complexity of modeling all these details into the user interests. In fact, by packing everything into a description, it may end up in very long sentences, made of several hundreds of tokens, which may make it difficult to process and understand. These two problems entail a complex task involving both scene understanding capabilities and cross-modal understanding.

3. Related work

There are several fields which are related to the proposed text-to-apartment recommendation problem. First of all, Section 3.1 will discuss the available datasets about apartments, and highlight the motivations which led us to collect a new dataset. Then, Section 3.2 will cover recent advancements in scene and language understanding. Finally, in Section 3.3 we explore the literature on the topic of cross-modal retrieval and contextualize our work into it.

3.1. About the dataset options for apartments

The proposed problem involves reasoning on both apartments and textual data. In Table 1 we summarize several characteristics of the publicly available datasets related to apartments or, more in general, house-like environments. Overall, we looked for a large-scale publicly available dataset containing indoor visual information (e.g., multiple photos or a 3d reconstruction), accompanied by descriptions of the contents, furniture, etc. However, most of the previously published datasets present some shortcomings which limit their applicability to our use case. Houses3k [32], HouseExpo [27], and Text-to-

3dHouseModel [14] offer only floorplan representation of the houses, which may not effectively capture appliances and other furniture, thus limiting their usefulness for recommending an apartment. FutureHouse [28], Matterport3d [11], and Habitat-Matterport3d [34] present high-quality panoramic, 3d scenes, images, or other visual formats, yet no language annotations are available. Closest to our work and task are SHREC 2018/2019 [2, 3, 46], House3d [44], and Rent3d [30]. However, both Rent3d and SHREC lack textual annotations, whereas House3d has some categorical information but it is not publicly available. Therefore, we decided to collect a new dataset, starting from the realistic rooms available in 3d-FRONT [22] and creating textual descriptions by aggregating the object, style, theme, and material information available in the dataset (details in Section 5.1). Noteworthily, this allows for learning to identify the furniture and appliances which are available, whereas the annotations in Text-to-3dHouseModel only described the type of rooms and their location within the house (i.e., not "detailed" with respect to the proposed task).

3.2. Scene understanding and language

Early works on scene understanding focused on analyzing the contents of 2d scenes and linking them to textual annotations, for instance when trying to automatize the generation of coherent descriptions [29], providing a correct answer for content-related questions [18], or dealing with text-guided indoor navigation [6]. Compared to the 2d counterpart, similar tasks are far less explored on 3d scenes. In the previous works, the focus was on associating language and single object instances. Chen et al. learned a joint 3d-text embedding space for text-guided generation purposes [13]. Achlioptas et al. focused on a similar objective, although working at a much finer granularity, learning to distinguish between very fine-grained details of the 3d objects [5]. Then, the research focused on more detailed descriptions, containing multiple objects and, possibly, rooms at the same time. This led to the creation of challenging tasks involving the identification of precise instances of an object among many distractors [4, 12] and, very recently, the automatic captioning of 3d scenes [16].

The task we propose in this paper is highly related to these advances. Nonetheless, its novelty and importance is highly motivating. First, current research fields are mostly related to localizing or generating objects, which are fundamental tasks yet they may not be as useful when trying to compare complex scenes and assess their relevance to the user interests. Second, the direct implications which our task can have both on businesses related to the metaverse, and on a more environmental-friendly approach to visiting apartments while looking for new accommodation.

3.3. Multimedia retrieval

Finding multimedia content which is relevant to the user interests among many hundreds of thousands or even millions of examples is a difficult problem which is getting more and more attention from the research community [23, 33]. The users commonly describe their interests by means of textual queries, which are then automatically mapped into a joint embedding space where another type of multimedia content is also mapped, e.g., images [33, 35] or videos [21, 23]. Recently, even more modalities were taken into consideration when building these spaces, e.g., by considering simultaneously text, image/video, and audio [7, 39]. The problem we are facing in this paper can be seen as retrieving apartments (i.e., scenes) by means of textual queries, as described in Section 2. This problem is interesting both for the challenges which we highlighted before, for its practical use cases, and also for its novelty. In fact, to the best of our knowledge, there are no previous works addressing the problem of *retrieving apartments*, or even more general 3d scenes, *by means of textual queries*. The only line of work about retrieving 3d scenes involves the use of images or sketches as queries [2, 3, 45]. However, the queries are visual and not textual, therefore limiting the usability of those approaches in our setting. Other works on 'scene retrieval' focus on locating the objects in the scene [31] or the text shown inside them [42, 43].

4. Proposed method: FArMARe

An overview of the proposed methodology, called Furniture-AwaRe Multi-task methodology for Apartment REcommendation (or FArMARe), is shown in Figure 2. The textual and visual data are processed by two independent branches, as typically done in cross-modal retrieval [10, 26, 33, 47]. In the following sections, we describe each of these branches separately, and how they are combined together.

4.1. Visual data

Starting from the 3d scene, multiple viewpoints are sampled, the amount of which depends on a subset of furniture categories present in the scene, as done in [17]. Each of the images is then processed by a pretrained Vision Transformer [20] to obtain a set of image descriptors. Then, our multi-task approach requires solving two tasks: ranking, and classification. For the first task, we use a simple network to learn a function which maps each image descriptor x_i into the joint embedding space:

$$\hat{x}_i = Conv1d(x_i) \tag{1}$$

$$a_i = ReLU(\hat{x}_i W_c + b_c) \tag{2}$$

where W_c and b_c are trainable parameters. At the same time, $\hat{x}_i$ is also processed by the network solving the sec-

Dataset	Data format	Textual annotations	Scale
Houses3k [32]	Outdoor/exterior	N. A.	Large
HouseExpo [27]	Floorplans	N. A.	Very large
Text-to-3dHouseModel [14]	Floorplans	Descriptions	Large
FutureHouse [28]	Indoor (panoramic)	N. A.	Very large
Matterport3d [11]	Indoor	N. A.	Large
HM3d [34]	Indoor	N. A.	Large
SHREC 2018/2019 [2, 3]	Indoor	N. A.	Large
House3d [44]	Indoor	Category per object/room	Large
Rent3d [30]	Floorplans and indoor	N. A.	Small
Ours	**Indoor**	**Detailed descriptions**	**Large**

Table 1: Comparison of several datasets from the literature on house models. "N. A." stands for "Not Available". Discussion in Section 3.1.

ond task, i.e., classification. The classifier learns to predict a class t_i for each of the images starting from the shared representation $\hat{x}_i$. To do so, it uses the following equations:

$$\begin{aligned} r_1 &= ReLU(\hat{x}_i W_{c_1} + b_{c_1}) \\ r_2 &= \delta(ReLU(r_1 W_{c_2} + b_{c_2})) \\ r_3 &= ReLU(r_2 W_{c_3} + b_{c_3}) \\ r_4 &= ReLU(r_3 W_{c_4} + b_{c_4}) \\ p_i &= softmax(r_4 W_{c_5} + b_{c_5}) \end{aligned} \tag{3}$$

where W_* and b_* are trainable parameters, and δ applies dropout with probability 0.1. The supervision for the classification is provided by automatically recognized furniture tags, leading to furniture-awareness in the training objective and in the features extracted in the intermediate layers. These tags are obtained by YOLOv8 [2] which we finetune on a dataset of furniture images [1] considering 25 different classes, e.g., 'bed', 'shelf', etc. In this way, the model needs to extract more general, furniture-aware features which are useful to solve both tasks simultaneously.

4.2. Textual data

As mentioned before, describing all the important aspects of an apartment may result in a very long description. Although the recent advancements in natural language processing made it easier to process and understand their contents [9, 15], very long sentences may still pose a problem [38]. To this end, we opted for the following approach. First of all, the apartment description is split into several sentences by splitting at the periods. Then, a pretrained Transformer [40] is used to extract several sentence-level representations, after which a bidirectional GRU is trained contrastively with the visual data. The final representation for the apartment description, d_i, is obtained by taking the mean of the final hidden state of the bidirectional GRU.

4.3. Loss function

The proposed approach, FArMARe, simultaneously learns to solve two tasks. For the first task, i.e., ranking, we use the triplet loss [36] which is commonly used in cross-modal retrieval settings [8, 21, 41]. Given a batch of N apartments, $a_1, \dots, a_N$, and their descriptions $d_1, \dots, d_N$, it can be defined as follows:

$$l_r(a,p,n) = \max(0, \Delta + s(a,n) - s(a,p))$$

$$\mathcal{L}_r = \frac{1}{2 \cdot N \cdot (N-1)} \Big(\sum_i \sum_{j \neq i} l_r(a_i, d_i, d_j) + \sum_i \sum_{j \neq i} l_r(d_i, a_i, a_j) \Big)$$

For the second task, i.e., classification, cross-entropy is used. It is defined as follows:

$$\mathcal{L}_c = -\frac{1}{N} \sum_{j=1}^{N} t_{i,j} \log(p_{i,j})$$

where $p_{i,j}$ is the softmax-normalized logit for the j-th class, and the tag t_i is seen as a one-hot representation where $t_{i,i} = 1$ and $t_{i,j} = 0$ for $i \neq j$. The final loss function is defined as:

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_c + \mathcal{L}_r)$$

5. Experimental results

The source code used in this paper for creating the dataset and training the models can be found on this Github.

5.1. Analysis of the dataset

The dataset used in this paper consists of 6081 indoor apartments which we gathered from 3d-FRONT [22]. Each apartment is annotated with a textual description which we collected by using the object categories, style, theme, and material annotations associated with each piece of furniture.

[2] https://yolov8.com/

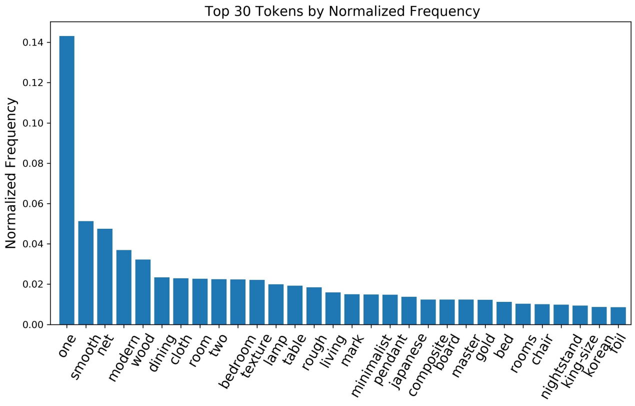


Figure 3: Top 30 most common tokens in the collected dataset. Discussion in Section 5.1.

On average, the descriptions have 319 words, ranging from a minimum of 19 and a maximum of 1905, making them very long; looking at the sentences, the average reduces to 16.4 (2 to 93), making them much easier to process. The words encountered in the annotations are highly specific to the task at hand, and describe several styles for the furniture (e.g., "Japanese" and "minimalist"), themes (e.g., "smooth" and "net"), and materials (e.g., "wooden" and "composite"). In Figure 3 we report the normalized frequency of the top 30 most common tokens, after the removal of stop-words and some words we use to create more human-like descriptions (e.g., "moreover", "additionally", etc). It can be seen that, apart from "one" which is common, the other tokens are not so frequent, making the retrieval task challenging, due to more difficulties in capturing relationships between words.

5.2. Experimental setting

We use two evaluation metrics to assess the performance of the approaches under analysis. The first metric, **recall@k** (or R@k), measures a model's ability to successfully retrieve the groundtruth information among the top k recommendations. Moreover, we also include the Rsum, which consists of the sum of R@1, R@5, and R@10 both for text-to-apartment and apartment-to-text. The second metric is the **median rank**, which reveals the position of the groundtruth in the ranked list. A lower median rank value indicates that the groundtruth is more likely to be found at higher positions, demonstrating better ranking accuracy.

In the following experiments, we both explore the use of a jointly trained vision and language backbone, and separately trained backbones. For the former, we consider a Vision Transformer, ViT-B-32 [20], and a 12-layer Transformer jointly trained via CLIP. The pretraining is done on LAION-2B (a subset of [37]). For the latter, we consider an ImageNet-pretrained ResNet-152 [24] and a pretrained BERT [19]. Is it important to note that in the non-learning baseline, only the jointly trained backbone is considered: in fact, it would not be possible to compute meaningful similarity scores by using ResNet and BERT, since the two share completely unrelated embedding spaces.

Here, we provide a brief description of the baseline methods. Overall, all the learning-based methods use sentence-level textual features followed by a bidirectional GRU for the text data modeling.

Non-learning baseline (NLB). In SHREC 2018/2019, which is the closest work to ours (see Section 3.1), non-learning based solutions were also proposed. The best-performing one used a pretrained VGG to extract features for both input data (i.e., the 2d query, and the 2d viewpoints of the 3d scene) and then computed the similarity matrix to perform the ranking. In our setting, this approach can not be used as is: in fact, our queries are textual and not visual, therefore imposing a domain gap too big to obtain a decent result. To address this and propose a non-learning baseline, we use a pretrained model jointly trained with CLIP [33] to extract both textual and visual features, separately pool them, and then compute the similarity matrix via cosine similarity. The model uses ViT-B-32 [20] for the visual backbone, and a 12-layer Transformer [40] for the textual one.

Average-pool based, FC network (AFN). To aggregate the visual descriptors x_i obtained from the backbone, this method simply averages them, obtaining $r_1 \in \mathbb{R}^{1\times F_V}$. Then, the following network is used to learn the final descriptor for the apartment, a_i:

$$\begin{aligned} r_2 &= BN(\delta_1(ReLU(W_{f_1}r_1 + b_{f_1}))) \\ r_3 &= BN(\delta_2(ReLU(W_{f_2}r_2 + b_{f_2}))) \\ a_i &= W_{f_3}r_3 + b_{f_3} \end{aligned}$$

where W_*, b_* are trainable weights and biases, $\delta_1(\cdot)$ and $\delta_2(\cdot)$ represent the dropout operator with probability 0.20 and 0.10, respectively. $BN(\cdot)$ identifies the use of Batch Normalization [25].

Conv1d network (CNV). Different from the previous method, the visual descriptors are not initially pooled. Instead, we use a simple network consisting of a 1d convolutional network to reduce the number of features, a ReLU activation function, and finally, a fully-connected layer to learn the final representation for the apartment.

5.3. Implementation details

For the 1d convolution used in Eq. 1 we use 512 output channels, stride one, and the kernel size of five. The classifier in Eq. 3 reduces the dimension by half in each layer, therefore going from 512 to 64. Then, we use the following 25 predefined classes: Bed, Cabinet, Carpet, Ceramic floor, Chair, Closet, Cupboard, Curtains, Dining Table, Door, Frame, Futec frame, Futec tiles, Gypsum Board,

Visual | *Textual*	Token-level	Sentence-level
ResNet | BERT	4.6	16.9
CLIP | CLIP	35.4	48.9

Table 2: Analysis of the performance (text-to-apartment R@10) when changing the input format for the textual annotations (long list of token-level features or shorter list of sentence-level features) and type of features (separately learned via ResNet-152 and BERT or jointly learned via CLIP). Discussion in Section 5.4.

Lamp, Nightstand, Shelf, Sideboard, Sofa, TV stand, Table, Transparent Closet, Wall Panel, Window, and Wooden floor. Moreover, we used a 26th class for those tags whose confidence was lower than 10%. To reduce the domain gap between the pretraining data and the data at hand, we fine-tuned YOLOv8 on 8000 images of furniture [1].

We use PyTorch 1.13.1 for the implementation and run all the experiments on a machine using an RTX A5000 GPU, 16 GB of RAM, and an Intel Xeon E5-1620. The training lasts 50 epochs with a batch size of 64. The optimizer is Adam and the learning rate starts from .008 and is decayed by a factor of 25% after 27 epochs.

5.4. Reasoning about the input data

The first two research questions we aim at answering are the following: (1) In the proposed setting, is it better to have the visual and textual backbones jointly or separately trained? (2) Considering that the descriptions feature hundreds of tokens, is it better if we extract the textual features at the token- or at the sentence-level?

To answer both questions, we perform the following experiment using the *AFN* method. For the jointly trained backbones, we consider ViT-B-32 and a Transformer trained with CLIP; for the other question, we use ImageNet-pretrained ResNet-152 [24] and BERT [19] to separately extract visual and textual features, respectively. We evaluate the four combinations (separate or joint features, token, or sentence level) and report the results in Table 2. To answer the first question, it can be seen that the features extracted from a jointly trained vision and language backbone lead to far better performance, achieving around 35.4% and 48.9% text-to-apartment R@10 at the token- and sentence-level respectively. These results represent a margin of +30.8% and +32.0%, respectively, compared to the separately trained features. For the second question, it can be seen that working on sentences makes the feature extraction process more effective, obtaining 16.9% R@10 using ResNet and BERT (+12.3% than the token-level), and 48.9% R@10 using CLIP (+13.5%).

5.5. Comparison between the baseline methods

In Table 3 we report a comprehensive quantitative analysis of the performance obtained by the three models we considered in our study (details in Section 5.2) and by our proposed method. As mentioned in the previous sections, it is not meaningful to use the non-learning baseline with ResNet and BERT, as the two present a wide domain gap between the two single-modal spaces. Then, from the Table, three main results are observed. First, all the methods under analysis achieve better performance when using the jointly trained backbones, as opposed to the separately trained ones, confirming the results obtained in the previous experiment. Second, delaying the fusion of the information obtained from the single viewpoints leads to better results than performing it early, since both CNV (e.g., 63.7 and 225.9 Rsum with ResNet and BERT, and CLIP features respectively) and FArMARe (92.3 and 232.7 Rsum) obtain far better performance than AFN (33.2 and 133.7 Rsum). Finally, the information obtained by the additional task in our multi-task approach leads to a considerable improvement over CNV, obtaining +1.1%, +2.0%, and +0.2% text-to-apartment R@1, R@5, and R@10, and overall +6.8 Rsum. Noteworthily, these results are obtained by performing 5 runs with both of the models, to reduce the influence of randomness.

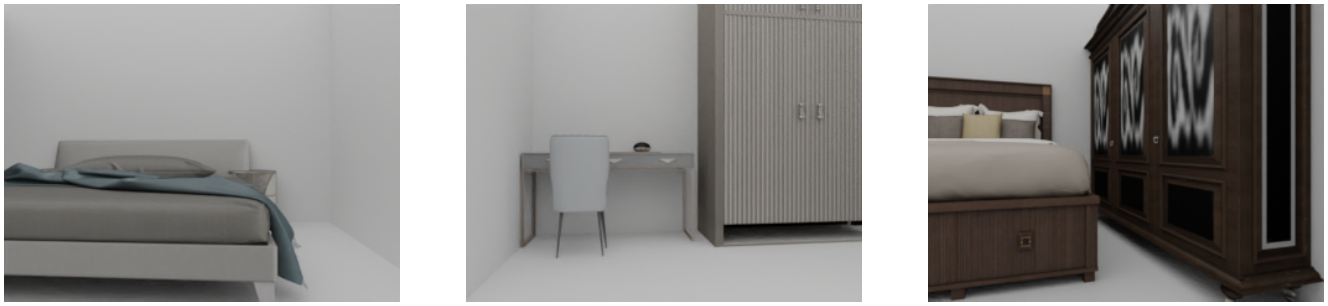
6. Discussion and limitations

About the object detection module. As mentioned in Section 4, we are currently extracting the best tag according to the detector. This works well when there is only one main object in the viewpoints (e.g., see Figure 4.a), however when multiple are visible picking only one tag may not be enough (e.g., see Figure 4.b). Moreover, the detection may also fail, leading to incorrect information injected into the visual descriptors (e.g., see Figure 4.c). While overall the proposed method proves to be effective, these limitations may indeed affect its final performance and reliability, leaving much space for improvement. We highlight the following options to address these shortcomings. First, considering that multiple object tags may be relevant for a single image, a different learning strategy could be designed to leverage this aspect (i.e., moving to a multi-label classification problem), while also adopting spatial attention to ground the predictions, making the approach both more explainable and possibly improving the overall performance. Second, in our setting, which consists of photorealistic yet synthetic environments, generating the supervision from synthetic data may help obtain a more precise classifier, which could also lead to better features for the ranking task.

About the long descriptions. In our methodology, we decided to split the descriptions into a sequence of sen-

Method	Text-to-Apartment				Apartment-to-Text				Rsum
	R@1	R@5	R@10	MedR	R@1	R@5	R@10	MedR	
ResNet | BERT features									
AFN	1.3	5.8	10.4	101	1.6	4.6	9.5	104	33.2
CNV	4.3	11.4	16.6	93	2.7	11.8	16.9	95	63.7
FArMARe (Ours)	**6.9**	**17.4**	**22.1**	**88**	**7.1**	**16.7**	**22.1**	**89**	**92.3**
CLIP features									
NLB (∼ [2])	0.1	0.6	1.3	413	0.6	1.9	3.7	339	8.2
AFN	10.6	25.5	34.2	29	8.9	23.2	31.3	31	133.7
CNV	23.7	40.6	48.9	**11**	23.1	40.7	48.9	12	225.9
FArMARe (Ours)	**24.8**	**42.6**	**49.1**	**11**	**25.7**	**41.4**	**49.1**	**11**	**232.7**

Table 3: Comparison between the baselines under analysis (Section 5.2), and the proposed method, FArMARe (Section 4). Discussion in Section 5.5.



(a) Tag: "bed"

(b) Tag: "wardrobe"

(c) Tag: "dining table"

Figure 4: Qualitative examples of the tags obtained from the object detector, including examples when (a) it correctly identifies the predominant furniture; (b) it correctly identifies one of the furniture, but misses the other; (c) it fails. Discussion in Section 6.

tences, then to capture the meaning of each of them separately by using a powerful language model, and finally to model their contextual information through a neural sequential model. By dividing the language understanding step into two sub-steps, we are able to achieve higher effectiveness. However, recent advancements showed that very long contexts can be understood by large language models [9, 15]. Therefore, by extending our methodology with these techniques it may be possible to obtain better language understanding even while working at the token-level.

7. Conclusion

In this paper, we defined the problem of recommending an apartment based on user-defined queries which capture their interests. This novel problem is inspired by the need for applications which are able to analyze the contents of a digital twin of the apartment, representing it in the metaverse, assess whether it is a good match for the user-defined query, and rank the metaverse apartments accordingly. This approach may be highly impactful both on businesses dealing with apartments recommendation, and on the environment, since being able to visit the apartment in the metaverse has a smaller impact than visiting it by physically moving across states.

To realize a model able of solving the problem, we introduced FArMARe, a multi-task methodology which uses both a contrastive framework, to learn how to perform cross-modal ranking, and a furniture-based classification objective, which helps the model extracting more general furniture-aware features. We collected and annotated a dataset of more than 6000 apartments, over which we performed several experiments with three different methods and two different feature extraction procedures, confirming the effectiveness of the proposed method.

Finally, we discussed the limitations of the model, while also highlighting possible future research directions.

Acknowledgements

This work was supported by the Department Strategic Plan (PSD) of the University of Udine–Interdepartmental Project on Artificial Intelligence (2020-25), by the Italian Ministry of University and Research (MUR)

Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) 2022 (project code 2022YTE579) and within the project DM737_HEU_voucher_2b_FALCON (CUP G25F21003390007), and by TechStar Srl, Italy.

References

[1] Mokhamed Nagy, https://universe.roboflow.com/mokhamed-nagy-u69zl/furniture-detection-qiufc/dataset/20. 5, 7

[2] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N Do, Heyu Zhou, Yang Zhou, et al. Shrec'18 track: 2d image-based 3d scene retrieval. *Training*, 700:70, 2018. 2, 4, 5, 8

[3] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Tobias Schreck, Ngoc-Minh Bui, Trong-Le Do, Mike Holenderski, Dmitri Jarnikov, Khiem T Le, et al. Shrec'19 track: Extended 2d scene image-based 3d scene retrieval. *Training (per class)*, 700:70, 2019. 2, 4, 5

[4] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 422–440. Springer, 2020. 4

[5] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947, 2019. 4

[6] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 4

[7] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 4

[8] Ayan Kumar Bhunia, Aneeshan Sain, Parth Hiren Shah, Animesh Gupta, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Adaptive fine-grained sketch-based image retrieval. In *European Conference on Computer Vision*, pages 163–181. Springer, 2022. 5

[9] Aydar Bulatov, Yuri Kuratov, and Mikhail S Burtsev. Scaling transformer to 1m tokens and beyond with rmt. *arXiv preprint arXiv:2304.11062*, 2023. 5, 8

[10] Min Cao, Shiping Li, Juntao Li, Liqiang Nie, and Min Zhang. Image-text retrieval: A survey on recent research and development. In *International Joint Conference on Artificial Intelligence (IJCAI-22) - Survey Track*, pages 5410–5417, 2022. 4

[11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE Computer Society, 2017. 4, 5

[12] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. 4

[13] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 100–116. Springer, 2019. 4

[14] Qi Chen, Qi Wu, Rui Tang, Yuhan Wang, Shuai Wang, and Mingkui Tan. Intelligent home 3d: Automatic 3d-house design from linguistic descriptions only. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12625–12634, 2020. 4, 5

[15] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023. 5, 8

[16] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3193–3203, 2021. 4

[17] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 4

[18] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018. 4

[19] Jacob Devlin, Ming-Wei Chang, and Kenton Lee. Google, kt, language, ai: Bert: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 6, 7

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 4, 6

[21] Alex Falcon, Giuseppe Serra, and Oswald Lanz. A feature-space multimodal data augmentation technique for text-video retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4385–4394, 2022. 4, 5

[22] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 4, 5

[23] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023. 4

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 7

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 6

[26] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7331–7341, 2021. 4

[27] Tingguang Li, Danny Ho, Chenming Li, Delong Zhu, Chaoqun Wang, and Max Q-H Meng. Houseexpo: A large-scale 2d indoor layout dataset for learning-based algorithms on mobile robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5839–5846. IEEE, 2020. 3, 5

[28] Zhen Li, Lingli Wang, Xiang Huang, Cihui Pan, and Jiaqi Yang. Phyir: Physics-based inverse rendering for panoramic indoor images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12713–12723, 2022. 4, 5

[29] Dahua Lin, Chen Kong, Sanja Fidler, and Raquel Urtasun. Generating multi-sentence lingual descriptions of indoor scenes. *arXiv preprint arXiv:1503.00064*, 2015. 4

[30] Chenxi Liu, Alexander G Schwing, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Rent3d: Floor-plan priors for monocular layout estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3413–3421, 2015. 4, 5

[31] Thao Nguyen, Nakul Gopalan, Roma Patel, Matt Corsaro, Ellie Pavlick, and Stefanie Tellex. Robot object retrieval with contextual natural language queries. *arXiv preprint arXiv:2006.13253*, 2020. 4

[32] Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Next-best view policy for 3d reconstruction. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 558–573. Springer, 2020. 3, 5

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4, 6

[34] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 4, 5

[35] Patsorn Sangkloy, Wittawat Jitkrittum, Diyi Yang, and James Hays. A sketch is worth a thousand words: Image retrieval with text and sketch. In *European Conference on Computer Vision*, pages 251–267. Springer, 2022. 3, 4

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 5

[37] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 6

[38] Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, et al. Scrolls: Standardized comparison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, 2022. 5

[39] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S Feris, David Harwath, James Glass, and Hilde Kuehne. Everything at once-multimodal fusion transformer for video retrieval. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 20020–20029, 2022. 4

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5, 6

[41] Chia-Hui Wang, Yu-Chee Tseng, Ting-Hui Chiang, and Yan-Ann Chen. Learning multi-scale representations with single-stream network for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6165–6175, 2023. 5

[42] Hao Wang, Xiang Bai, Mingkun Yang, Shenggao Zhu, Jing Wang, and Wenyu Liu. Scene text retrieval via joint text detection and similarity learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2021. 4

[43] Lilong Wen, Yingrong Wang, Dongxiang Zhang, and Gang Chen. Visual matching is enough for scene text retrieval. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 447–455, 2023. 4

[44] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. In *ICLR Workshop 2018*, 2018. 4, 5

[45] Juefei Yuan, Hameed Abdul-Rashid, Bo Li, and Yijuan Lu. Sketch/image-based 3d scene retrieval: Benchmark, algorithm, evaluation. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 264–269. IEEE, 2019. 4

[46] Juefei Yuan, Hameed Abdul-Rashid, Bo Li, Yijuan Lu, Tobias Schreck, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N

Do, Trong-Le Do, et al. A comparison of methods for 3d scene shape retrieval. *Computer Vision and Image Understanding*, 201:103070, 2020. 4

[47] Cunjuan Zhu, Qi Jia, Wei Chen, Yanming Guo, and Yu Liu. Deep learning for video-text retrieval: a review. *International Journal of Multimedia Information Retrieval*, 12(1):3, 2023. 4