Fusing Domain-Specific Content from Large Language Models into Knowledge Graphs for Enhanced Zero Shot Object State Classification

Filippos Gouidis[1,2,3]**, Katerina Papantoniou**[3]**, Konstantinos Papoutsakis**[2] **Theodore Patkos**[3]**, Antonis Argyros**[2,3] **and Dimitris Plexousakis**[2,3]

[1]Computer Science Department, University of Crete, Heraklion, Greece
[2]Department of Management, Science and Technology, Hellenic Mediterranean University, Agios Nikolaos, Greece
[3]Institute of Computer Science, Foundation for Research and Technology Hellas, Heraklion, Greece
gouidis@{ics.forth.gr, csd.uoc.gr, hmu.gr}, kpapoutsakis@hmu.gr, {papanton, patkos, argyros, dp}@ics.forth.gr

Abstract

Domain-specific knowledge can significantly contribute to addressing a wide variety of vision tasks. However, the generation of such knowledge entails considerable human labor and time costs. This study investigates the potential of Large Language Models (LLMs) in generating and providing domain-specific information through semantic embeddings. To achieve this, an LLM is integrated into a pipeline that utilizes Knowledge Graphs and pre-trained semantic vectors in the context of the Vision-based Zero-shot Object State Classification task. We thoroughly examine the behavior of the LLM through an extensive ablation study. Our findings reveal that the integration of LLM-based embeddings, in combination with general-purpose pre-trained embeddings, leads to substantial performance improvements. Drawing insights from this ablation study, we conduct a comparative analysis against competing models, thereby highlighting the state-of-the-art performance achieved by the proposed approach.

Introduction

The deep learning paradigm has revolutionized the standard approach concerning AI-based problem-solving in several scientific disciplines over the last few years, leading to spectacular breakthroughs. However, the growing utilization and customization of these types of methods brought to light several limitations, the most important of which are (a) their total dependency on significant quantities of annotated data whose collection and annotation is a costly and time-consuming procedure, and (b) the vast computational cost entailed by these approaches. Therefore, a great deal of research has been dedicated in recent years to the development of methods to overcome these limitations.

General-purpose semantic representations, typically in the form of pre-trained word vectors, can be considered as an answer to these challenges since they offer several attractive features. First, they are readily available, straightforward to use, and can be easily integrated in many different frameworks that address a wide variety of tasks. Equally important, in certain cases, performance gains can be obtained by their utilization, especially when used in conjunction with Knowledge Graphs (KGs) which enable a structured and robust way to represent information concerning entities and

relationships, and serve as powerful tools for organizing and connecting diverse knowledge in a networked fashion. Nevertheless, even though these general-purpose representations are often trained on vast corpora, they still lack coverage for specialized topics, while it also remains difficult to handle polysemy. Such representations can provide an average representation based on the different meanings of a word, but this may not suffice in specific problems.

On the other hand, domain-specific knowledge can mitigate the aforementioned limitations. However, the generation of such knowledge requires typically human expertise and considerable amounts of time and effort. The advent of Large Language Models (LLMs) seems to offer an efficient way to overcome these hurdles since models of this type can provide expertise comparable to that of humans in a fast and semi-automated way. The utilization of LLMs poses other challenges, the most stressing of which concern their effective prompting, their struggle with commonsense knowledge and reasoning, and their difficult discernible biases (Sartori and Orrù 2023).

In light of these points, significant research in various fields is currently concentrated on performing prediction tasks on samples for which no prior training has been performed, an approach that is typically referred to as zero-shot learning. Computer Vision is one of the fields, where zero-shot methods are extensively investigated. One reason for this is the already mentioned cost for data collection and annotation. Another important advantage of the zero-shot-based methods is that standard methods, when utilized in datasets with different distributions from the ones that have been trained, typically suffer from performance degradation. However, zero-shot methods are impervious to this kind of limitation. Nevertheless, the development of robust zero-shot methods presents a formidable challenge with its own set of unique hurdles, the most important of which are the *retrieval* of pertinent information that will be used by the models as a proxy to the "missing" visual information and the effective *projection* of this information to the visual space, enabling its utilization for the associated vision task.

In this study, we consider the problem of vision-based Object State Classification (OSC), a task involving the recognition of object states in images. Efficient OSC is crucial for any agent interacting with objects, given the profound impact states can exert on overall object functional-

ity. OSC presents several significant challenges that remain unresolved and is currently a focal point of increasing interest within the research community. Specifically, our attention is directed toward the challenging variant of Object-Agnostic zero-shot OSC, which aims to deduce the state of an object without relying on visual information related to the state classes or any general information about object classes, as described in (Gouidis et al. 2024). Building upon recent research addressing the specified problem (Nayak, Yu, and Bach 2022; Gouidis et al. 2024), we extend the state-of-the-art by integrating LLM frameworks into the broader pipeline designed for addressing Object-Agnostic zero-shot OSC, investigating appropriate schemes for an efficient coupling of the components. A key feature of our methodology is the utilization of an LLM that generates domain-specific knowledge relevant to the task.

To investigate this approach, we conduct a thorough ablation study to explore the optimal way in which the information derived from the LLM can be effectively leveraged. The experimental results indicate that the fusion of this domain-specific knowledge with pre-trained general-purpose knowledge has the potential to achieve state-of-the-art (SoA) performance in the Object-Agnostic zero-shot OSC task.

We believe that the insights gained from our approach extend beyond the specific problem under examination and can prove advantageous for other challenges associated with zero-shot learning. Additionally, it is essential to emphasize that our work represents an initial stride towards creating an effective framework that seamlessly integrates Large Language Models (LLMs) into a hybrid method, incorporating both low-level (data-driven) and high-level (symbolic) components.

The preliminary results presented in this work instill confidence in the potential of this approach. We view it as a promising step towards the development of an efficient framework that can serve as a foundation for addressing a broader spectrum of problems, particularly those within the realm of zero-shot learning.

In summary, this paper makes the following contributions:

- We introduce a novel approach that integrates LLMs into the successful zero-shot framework for object state classification which combines Knowledge Graphs and pre-trained semantic vectors.

- We present a semi-automatic method that harnesses the general-purpose content of an LLM to generate domain-specific knowledge.

- Through an extensive ablation study, we explore various aspects of the optimal integration of our method.

- Building upon the insights gained from the ablation study, we compare our method against competing models, demonstrating its state-of-the-art performance.

Related Work

Leverage LLMs: The integration of LLMs into tasks involving KGs embedding is an active area of research, as demonstrated by (Pan et al. 2023). (Wang et al. 2021) introduces KEPLER, a unified model for knowledge embedding and pre-trained language representation. This model encodes textual entity descriptions into embeddings using a pre-trained model, achieving joint training objectives for knowledge embedding (KE) and masked language modeling (MLM). In another study, (Huang et al. 2022) combines LLMs with vision and graph encoders to generate multimodal KG embeddings. Evaluation on multilingual named entity recognition and visual sense disambiguation tasks indicates improved performance compared to alternative approaches. LLMs have also been applied to assist open-vocabulary models for zero-shot image classification, as seen in the work of Pratt, Liu, and Farhadi (2022). This approach involves constructing prompts specialized for the target dataset for the GPT-3 model, with the goal of generating text containing crucial discriminating characteristics of image categories.

Fusion of word embeddings: Various approaches for the fusion of embeddings in Natural Language Processing have been explored. In (Ghannay et al. 2016), a comprehensive evaluation of different word embeddings (e.g., CBOW, Glove, Skip-gram, w2vf-deps) demonstrates performance improvements through their combination, emphasizing complementarity. The fusion of generic and domain embeddings, closely aligned with the objective of this work, is investigated in (Sarma, Liang, and Sethares 2018) and (Seyler and Zhai 2020). In (Sarma, Liang, and Sethares 2018), an approach aligns word vectors using Canonical Correlation Analysis and non-linear Kernel Canonical Correlation Analysis (Fukumizu, Bach, and Gretton 2007), followed by combining generic and domain embeddings through convex optimization. (Seyler and Zhai 2020) explores three fusion approaches: concatenation of embedding vectors, weighted fusion of text data, and interpolation of aligned embedding vectors, with the interpolation method yielding optimal results. Moreover, (Rettig, Audiffren, and Cudre-Mauroux 2019) suggests relying on domain-only embeddings for specific applications, proposing a two-step process. The first step involves ranking to capture the similarity between the downstream application corpus and various domain-specific embeddings. The second step employs fusion-based approaches, with Principal Components Analysis (PCA) over concatenated embeddings identified as the most efficient method.

State Classification: Few works address state classification specifically. The majority of the works related to the task adopts assumptions akin to those used in attribute classification tasks (Gouidis et al. 2022). A significant research focus in this domain is zero-shot learning, gaining attention for its practical relevance in real-world applications, especially in overcoming challenges associated with collecting and learning training data for a large number of object classes (Xian et al. 2018a). An influential approach in zero-shot learning involves using semantic embeddings to represent objects and their attributes in a low-dimensional space (Wang, Ye, and Gupta 2018). The emergence of powerful generative models has opened up a promising avenue for zero-shot object classification (Xian et al. 2018b). In a more recent work, (Gouidis et al. 2024) specifically tackles the challenging task of Object-Agnostic zero-shot State Classification, employ-

ing a strategy that leverages Knowledge Graphs and pre-trained semantic vectors.

Vision-Language Models: Vision-Language Models (VLMs) are sophisticated language models that harness contrastive learning to bridge the gap between images and text. They are essentially an extension of LLMs, trained on massive text data, tailored to tackle Computer Vision challenges. VLMs achieve this by jointly training image and text encoders on vast datasets of image-text pairs gleaned from the internet. This synergistic approach empowers the encoders to excel in downstream tasks like Image Captioning, Visual Question Answering, and Zero-Shot Classification. Notable examples of VLMs include CLIP (Radford et al. 2021), ALIGN (Jia et al. 2021) and BLIP (Li et al. 2022).

Methodology

Given a set of objects, O, a set of states, S, and a set of images, I, with each image, $i \in I$ containing an object $o \in O$ that is in a state $s \in S$, the goal of OSC is to predict the state label $s \in S$, given an object o appearing in an image $i \in I$ as input. In the zero-shot variation of OSC that we examine in this work, no images are used for the training of the classifier model that is deployed for the prediction.

Approach

Our approach consists of the following stages which are graphically illustrated in Figure 1:

1. Prompting of the LLM and generation of a text corpus.

2. Construction of a KG relevant to the classes we are interested in recognizing.

3. Production of semantic and visual embeddings.

4. Training of a GNN that learns to project semantic embeddings into visual space.

5. Projection of the semantic embeddings of stage 3 to the visual space via the utilization of the KG of step 2, the visual embeddings of stage 3 and the trained GNN of stage 4.

6. Incorporation of the projected embeddings of stage 5 into a pre-trained Visual Classifier.

Stage 1: Prompting & semantic representation. Prompting: The prompts[1] that were directed to the LLM can be roughly categorized into two groups. The first group contains prompts that examine the various objects as items of the physical world and emphasize on the description of their properties. The second group focuses on the visual characteristics of the objects and how they are depicted in images or described in image captions.

More specifically, for the first group some of the prompts are related only to the classes of ImagetNet (1K version) e.g., *'For the concept [ImageNet class] give me a short narration of about 160 words'*. Other prompts combine the classes of the ImageNet with a set of predefined states, e.g.,

---

[1]The prompts are available at https://docs.google.com/spreadsheets/d/11jo4wSdBrI-rj3jI-UpU39KDfBGGDhsxIEs9UUShh1w/edit?usp=sharing

Corpus	Total	Unique	Vocab
Basic	665.594	13.688	7.691
Extended	1.573.843	19.183	11.372

Table 1: Overview of the two corpora utilized in our work w.r.t. number of total, unique and vocab tokens. The vocabulary for both corpora consists of unique tokens with a minimum frequency equal to 3.

'make a connection between the concept [ImageNet class] and [state1, ..., stateN]', while others try to find asymmetries e.g., *'What action can I do with an object in state [state] in the physical world that I cannot do when the object is in [opposite state]?'* and causal chains between pair of states e.g., *'Give me 10 examples of how a human can take an [state] object of the physical world and make it [opposite state]'*.

Most of the prompts follow a zero-shot prompting style but in-context learning in the form of few-shot prompting e.g., *'What objects of the physical world other than charger, phone, socket can be also connected?'* and chain-of-thought via the simplistic addition of *'Let's consider it step by step.'* at the end of the prompts, have also applied. The second group consists of prompts like *'Describe what a(n) [ImageNet class] looks like?'*, *'Describe a photo of a(n) [ImageNet class]'*, and *'Give me a caption of an image of a(n) [ImageNet class]'*.

Corpus processing: Initially, the responses of the model were cleaned up from the text that started the conversation. In most of the cases, the text follows a standardized format e.g., *'As a helpful and respectful assistant, I would be happy to assist you'*, *'Sure, I'd be happy to help!'* etc. Table 1 shows an overview of the corpora. The first corpus (Basic) corresponds to the first group of prompts while the second (Extended) to the unification of the two groups of prompts.

LLM-based and generic embeddings: For the training of the LLM-based static word embeddings (domain) the Global Vectors (GloVe) (Pennington, Socher, and Manning 2014), the fastText (Bojanowski et al. 2017) and the word2vec (Mikolov et al. 2013) algorithms were used. As general-purpose embeddings, we employed pre-trained word embeddings of the GloVe[2], the fastText[3] and the word2vec[4] models. In addition, the ConceptNet Numberbatch[5] KG-enhanced word embeddings were also used which retrofit (Faruqui et al. 2014) word2vec and GloVe word representations to the ConceptNet graph (Speer, Chin, and Havasi 2017).

Each type of embeddings was employed both separately and in conjunction with other embeddings types. For the combination of the embeddings of different type we experimented with a number of different fusion techniques. The goal was to exploit both the robustness of the generic embeddings and the domain-specific semantic information that is

---

[2]https://github.com/stanfordnlp/GloVe (6B and 42B versions)

[3]https://fasttext.cc/docs/en/english-vectors.html

[4]https://github.com/harmanpreet93/load-word2vec-google

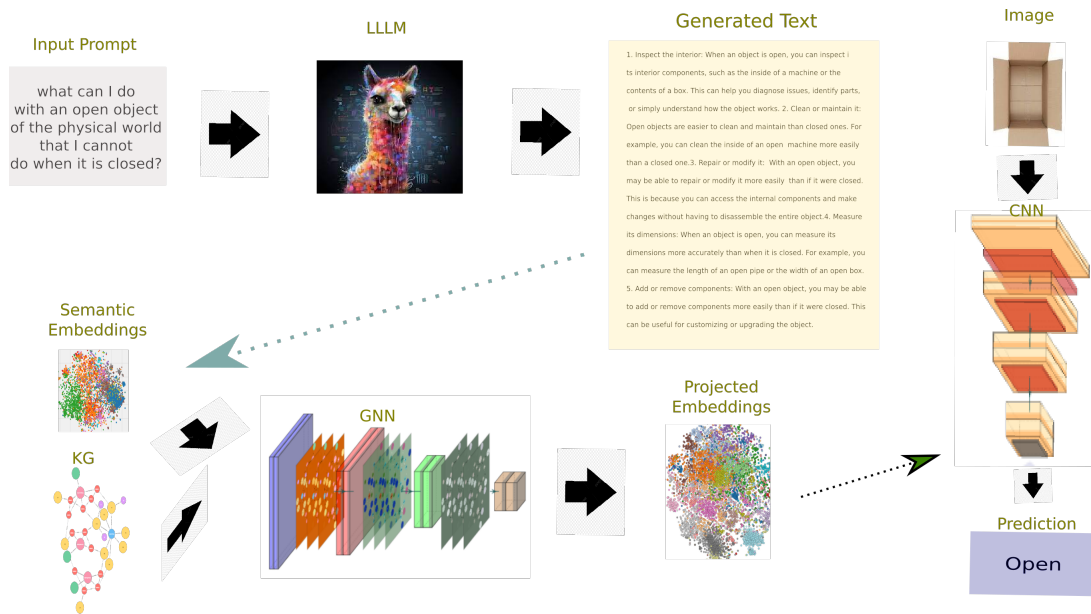[5]https://github.com/commonsense/conceptnet-numberbatch

Figure 1: The schematic representation of our methodology. A Large Language Model (LLM) is given specific prompts associated with the target classes we aim to identify. The resulting corpus is processed, leading to the generation of semantic embeddings. Subsequently, these vectors are fed into a Graph Neural Network (GNN), previously trained to map embeddings from the semantic space to the visual space. The resulting visual embeddings are then integrated into the final layer of a pre-trained Convolutional Neural Network (CNN) classifier.

captured by the LLM-based embeddings. Below, we briefly describe the undertaken fusion approaches:

- **Averaging:** The arithmetic mean of the two-word embedding sets is calculated following Coates and Bollegala (2018).

- **Concatenation:** The LLM-based embeddings were concatenated with the generic embeddings for the common words between the two vocabularies. The intuition for this setting is that the model during training will prioritize certain embedding dimensions.

- **PCA:** The concatenated embeddings were linearly transformed via the PCA technique to preserve the relevant information by approximating the input vectors and minimizing the number of components. A z-means normalization was applied beforehand and we kept the top-300 components.

- **UMAP:** In this case, the UMAP (McInnes et al. 2018) which is a non-linear dimension reduction technique, was employed over the concatenated embeddings. UMAP has previously been applied for dimensionality reduction on text embeddings (Massarenti and Lazzarinetti 2021).

- **Autoencoder:** An auto-encoder (Vincent et al. 2008) was employed taking as input the concatenated embeddings. For each word, we consider the vector of values produced by the latent layer as the combined word embedding.

Stage 2: KG construction. In this stage, a Knowledge Graph is built, with its topology tailored to the target classes. The process involves querying one or more commonsense knowledge sources to retrieve neighbors for each target class, up to a specified hop threshold. The KG is then constructed by merging all query results. Intentionally, no filtering or other post-processing has been performed on the query results, to ensure a rapid and automatic execution of the entire stage. As such, it is important to note that this approach unavoidably introduces noise into the constructed KG.

Stage 3: Production of Semantic and Visual Embeddings. The corpus generated by the LLM in the initial stage undergoes processing, leading to the generation of semantic representations in the form of word embeddings. These domain-specific embeddings become pivotal in subsequent stages, either utilized in isolation or in conjunction with general-purpose pre-trained embeddings. To generate visual embeddings, a Convolutional Neural Network (CNN) classifier is trained on images, and the last layer's learned weights serve as visual embeddings corresponding to the classes of the images used for training.

Stage 4: GNN Training. A Graph Neural Network (GNN) is trained to effectively project semantic embeddings onto the visual space. Specifically, the GNN takes as input a KG and semantic embeddings corresponding to each node in the KG. The training procedure involves learning to generate visual embeddings for each node. The visual embeddings used to guide the GNN training are those obtained in stage 3. The significance of the GNN lies in its ability to capture graph dependencies through message passing between nodes, as elucidated by (Zhou et al. 2020). With this approach, the GNN is designed to learn the interrelations between nodes

Combination	CN	VG	WN	WK	AT
KG1	✓				
KG2		✓			
KG3	✓	✓			
KG4	✓		✓		
KG5	✓			✓	
KG6	✓		✓	✓	
KG7		✓	✓		
KG8		✓		✓	
KG9		✓	✓	✓	
KG10	✓	✓	✓	✓	✓

Table 2: Details of the KGs used for the production of the visual embeddings (stage 5 of methodology). Each KG combination is constructed for three different hops thresholds: 1,2 and 3. CN: ConceptNet, VG: Visual Genome. WN: WordNet, WK: WikiData, AT: Atomic.

in KGs. In our scenario, this involves the KG that is provided as input to the GNN during the training process and the KGs that are provided during the embedding projection phase that is described next.

Stage 5: Embeddings Projection to the Visual Space. The GNN, trained in stage 4, is fed with the KG constructed in stage 2 and the semantic embeddings generated during stage 3. The GNN processes this input and produces visual embeddings corresponding to the target classes. This step ensures that the GNN effectively leverages both the structure of the KG and the semantic information encoded in the embeddings to generate meaningful visual representations for the specified target classes.

Stage 6: Zero-shot classifier adaptation. In the final step, the last layer of the CNN classifier, initially utilized as a guide in stage 3, is replaced with the visual embeddings generated in the previous stage. This adaptation enables the modified CNN to effectively discern and identify the target classes, leveraging the refined visual embeddings for improved classification performance.

Experimental Evaluation

We conducted a series of experiments with a dual objective. Our primary goal was to explore crucial aspects related to the integration of LLMs into the Object Agnostic Zero Shot Classification task pipeline. To achieve this, we carried out a comprehensive ablation study covering various parameters. Additionally, we aimed to evaluate the performance of the best-ablated model against existing competing methods. In what follows, we provide detailed insights into the implementation and results for both experimental settings.

Implementation & Evaluation Issues

The LLM utilized in our study is Llama2 (Touvron et al. 2023), an updated version of LLama1. Llama2 belongs to the family of decoder-only large language models, comprising a suite of pre-trained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters.

The family of Llama2 models is freely accessible for both research and commercial purposes[6]. For the purposes of this research, we specifically employed the 13 billion parameters variant of the Llama-2-Chat model. In particular, a quantized model provided by TheBloke[7] was utilized. The selection of the Llama 2-Chat model was made based on its generally superior performance compared to other available open-source chat models. The predefined values for temperature and top_p parameters were maintained at 0.6 and 0.9, respectively.

trim figure

The KG utilized for the projection from semantic space to visual space is derived from the ImageNet Graph, organized according to the WordNet hierarchy. This KG corresponds to the ImageNet1K dataset, comprising approximately 1,350,000 images across 1000 different classes (Krizhevsky, Sutskever, and Hinton 2012). The GNN model is trained from scratch, following the methodology outlined in (Kampffmeyer et al. 2019), for 1500 epochs. The architecture of the network is convolutional-based and is inspired by (Nayak, Yu, and Bach 2022). During training, 950 classes were randomly selected from the ImageNet (ILSVRC 2012) dataset (Russakovsky et al. 2015), while the remaining 50 classes were reserved for validation purposes. Two widely used pre-trained classifiers, namely ResNet-101 (Gong et al. 2022) and ViT-B/16 (Dosovitskiy et al. 2021), are employed in our experiments.

For the generation of visual embeddings, five commonsense sources were queried to construct KGs: ConceptNet (Speer, Chin, and Havasi 2017), WordNet (Miller 1994), Visual Genome (Krishna et al. 2017), Wikidata (Vrandečić and Krötzsch 2014), and Atomic (Sap et al. 2019). In total, 30 different KGs were constructed, representing 10 combinations of sources across three hop distances: 1, 2, and 3. Table 2 provides detailed information about these KGs.

Concerning semantic embeddings, we experimented with four different approaches for semantic representation: Glove, Word2Vec, fastText, and Numberbatch. For both types of embeddings (domain and general purpose) a dimension (d) of 300 is used, which is a common choice for word embeddings. In addition, previous work (Lai et al. 2016) has shown that the performance of embeddings tends to plateau after a certain dimension i.e., 100. In respect to the preparation of corpus for the training of word embeddings, the segmentation of the sentences, and the tokenization was performed with the StanfordCoreNLP (Qi et al. 2018) and all tokens were lowercased.

Datasets: Except for the OSDD dataset (Gouidis et al. 2022), which is explicitly designed for the state detection task, currently, no other dataset exists exclusively dedicated to capturing object states in images. Nonetheless, certain pre-existing datasets focused on object detection and classification, incorporate object states as a subset within their array of object classes. One such dataset is the Visual Attributes in the Wild (VAW) dataset (Pham et al. 2021), en-

---

[6] https://about.fb.com/news/2023/07/llama-2
[7] The llama-2-13b.Q5-K-M.gguf file at https://huggingface.co/TheBloke/Llama-2-13B-GGUF

Corpus	OSDD	CGQA-States	MIT-States	VAW
Bas Domain-No	19.60 ± 2.00	31.07 ± 3.51	29.54 ± 2.90	17.93 ± 2.88
Bas Domain	$\mathbf{24.64 \pm 2.01}$	36.32 ± 2.23	$\mathbf{37.80 \pm 3.68}$	$\mathbf{24.52 \pm 3.13}$
Ext Domain	18.83 ± 1.52	$\mathbf{38.59 \pm 4.33}$	36.53 ± 5.00	19.79 ± 1.26

Table 3: Results for different corpora. Bas Domain-No: Basic corpus without stopwords. Bas Domain: Basic corpus with stopwords. Ext Domain: Extended corpus with stopwords. In all cases, the GloVe representation was employed as the chosen method for encoding the textual information within the respective corpus.

Semantic Source	OSDD	CGQA-States	MIT-States	VAW
GloVe 6B	23.55 ± 0.61	37.54 ± 3.07	$\mathbf{46.58 \pm 4.17}$	$\mathbf{25.45 \pm 2.54}$
GloVe 42B	21.97 ± 2.78	37.81 ± 4.61	41.78 ± 3.96	17.97 ± 3.21
fastText 16B	23.42 ± 1.90	$\mathbf{40.12 \pm 4.31}$	42.67 ± 2.75	23.64 ± 1.97
Word2Vec	23.42 ± 1.90	40.12 ± 4.31	42.67 ± 2.75	23.64 ± 1.97
Numberbatch	$\mathbf{24.29 \pm 1.35}$	26.56 ± 2.19	36.48 ± 2.60	19.08 ± 2.39
GloVe Bas Domain	$\mathbf{24.64 \pm 2.01}$	36.32 ± 2.23	37.80 ± 3.68	$\mathbf{24.52 \pm 3.13}$
fastText Bas Domain	23.41 ± 2.61	$\mathbf{41.54 \pm 2.57}$	$\mathbf{40.92 \pm 2.41}$	24.23 ± 2.84
Word2Vec Bas Domain	24.11 ± 2.25	42.01 ± 2.51	42.05 ± 2.61	24.21 ± 2.75

Table 4: Results for different types of semantic embeddings. GloVe 6B: 6B tokens trained on Wikipedia 2014 and Gigaword 5. GloVe 42B: 42B tokens trained on Common Crawl. fastText 16B: 16B tokens trained on Wikipedia 2017, UMBC webBase corpus, and statmt.org news dataset. Word2Vec: 100B tokens trained on a part of the Google News dataset. Numberbatch: retrofitted word2vec-100B trained on a part of the Google News dataset and GloVe-840B trained on Common Crawl word representations to the ConceptNet 5.5 graph.

Semantic Source	OSDD	CGQA-States	MIT-States	VAW
GloVe 6B	23.55 ± 0.61	37.54 ± 3.07	$\mathbf{46.58 \pm 4.17}$	25.45 ± 2.54
GloVe 6B + Bas Domain	24.77 ± 2.27	42.24 ± 2.93	42.58 ± 2.35	25.06 ± 2.61
GloVe 6B + Ext Domain	$\mathbf{25.23 \pm 2.54}$	$\mathbf{43.11 \pm 4.23}$	42.43 ± 4.62	$\mathbf{25.97 \pm 2.19}$
GloVe 6B + Ran	21.51 ± 1.22	29.78 ± 4.59	43.93 ± 3.00	20.95 ± 1.90
GloVe 6B - BDCVR	23.17 ± 1.73	37.30 ± 3.97	43.58 ± 2.23	22.64 ± 1.44
GloVe 42B	21.97 ± 2.78	37.81 ± 4.61	41.78 ± 3.96	17.97 ± 3.21
GloVe 42B + Bas Domain	23.11 ± 2.93	39.31 ± 4.54	42.96 ± 4.14	20.14 ± 3.74
GloVe 42B + Ext Domain	$\mathbf{23.64 \pm 2.78}$	$\mathbf{40.47 \pm 4.59}$	$\mathbf{43.92 \pm 4.35}$	$\mathbf{22.31 \pm 3.54}$
GloVe 42B + Ran	19.54 ± 2.14	30.14 ± 2.84	38.54 ± 4.02	15.56 ± 2.74

Table 5: Results for different concatenation combinations of domain-specific and general-purpose embeddings. In all cases, the concatenated embeddings have a dimensionality of 600. Ran: Random Glove embeddings. BDCVR: The general-purpose embeddings have the same coverage as Bas Domain embeddings. For explanations regarding the remaining abbreviations, please refer to the previous tables.

Method		OSDD	CGQA-States	MIT-States	VAW
	AVG	23.91 ± 1.65	37.88 ± 3.23	41.42 ± 3.50	23.19 ± 1.73
	Concat	24.77 ± 2.27	$\mathbf{42.24 \pm 2.93}$	42.58 ± 2.35	25.06 ± 2.61
GloVe 6B + Bas Domain	PCA	$\mathbf{27.27 \pm 1.23}$	37.45 ± 3.54	$\mathbf{45.22 \pm 6.50}$	25.15 ± 3.15
	UMAP	25.14 ± 1.58	34.86 ± 3.06	30.06 ± 2.84	$\mathbf{25.32 \pm 2.80}$
	Autoencoder	24.69 ± 1.82	38.77 ± 4.20	42.64 ± 5.10	24.57 ± 2.55
	AVG	24.33 ± 1.82	39.08 ± 3.42	42.87 ± 3.47	23.54 ± 1.92
	Concat	25.68 ± 2.65	40.12 ± 2.47	41.35 ± 2.11	27.14 ± 2.61
GloVe 6B + Ext Domain	PCA	26.56 ± 2.90	$\mathbf{42.94 \pm 4.07}$	42.08 ± 4.57	25.33 ± 2.69
	UMAP	27.41 ± 1.78	36.41 ± 3.42	32.81 ± 2.79	26.49 ± 2.30
	Autoencoder	$\mathbf{28.55 \pm 1.46}$	41.53 ± 4.45	$\mathbf{49.02 \pm 3.13}$	$\mathbf{29.29 \pm 3.64}$

Table 6: Results for different fusion methods of domain-specific and general-purpose embeddings. AVG: The two embeddings were averaged. Concat: The two types of embeddings were concatenated resulting in vectors of dimension 600. PCA: Principal Component Analysis is performed on the concatenated embeddings, producing a vector of dimension 300. UMAP: Uniform Manifold Approximation and Projection is performed to the concatenation of the two embeddings, producing a vector of dimension 300. Autoencoder: An autoencoder is employed, with the concatenated embeddings serving as input and a vector of dimension 300 is produced. For explanations regarding the remaining abbreviations, please refer to the previous tables.

Sources/Technique	CNN Model	OSDD	CGQA-States	MIT-States	VAW
Bas Domain	RN101	24.64 ± 2.01	**36.32 ± 2.23**	**37.80 ± 3.68**	24.52 ± 3.13
	ViT-16	**25.26 ± 1.94**	34.91 ± 6.40	32.03 ± 2.88	**26.75 ± 3.99**
GloVe 6B + Bas Domain (PCA)	RN101	**27.27 ± 1.23**	**37.45 ± 3.54**	**45.22 ± 6.50**	**25.15 ± 3.15**
	ViT-16 (PCA)	25.63 ± 1.59	30.08 ± 2.94	39.24 ± 4.30	23.09 ± 1.41
GloVe 6B + Bas Domain (UMAP)	RN101	25.14 ± 1.58	**34.86 ± 3.06**	**30.06 ± 2.84**	25.32 ± 2.80
	ViT-16	**27.09 ± 2.09**	30.65 ± 2.39	24.84 ± 2.28	**25.75 ± 2.53**
GloVe 6B + Ext Domain (PCA)	RN101	26.56 ± 2.90	**42.94 ± 4.07**	**42.08 ± 4.57**	**25.33 ± 2.69**
	ViT-16 (PCA-Aug)	**27.14 ± 1.26**	35.01 ± 2.30	35.10 ± 3.70	**25.33 ± 2.57**

Table 7: Results for the two CNN Models that are used in the final stage of classification and from which the visual embeddings are extracted in stage 2. RN101: ResNet-101. ViT-16: Vit-16/B. For an explanation regarding the remaining abbreviations please refer to previous tables.

Method	OSDD	CGQA-States	MIT-States	VAW
Ours	**33.0**	**48.1**	**53.7**	**32.7**
OaSc	29.1	39.4	47.4	25.4
CLIP-RN101	22.5	46.9	39.3	28.0
CLIP-VITBP16	28.8	44.9	46.4	30.1
CLIP-VITLP14	28.4	43.4	48.6	27.9
ALIGN	29.5	40.0	44.2	28.4
BLIP	13.3	26.0	27.2	16.1
RN101 (Supervised)	67.5	60.5	85.3	51.9

Table 8: Experimental results of the proposed approach for the zero-shot object state classification task. The performance reported refers to the following four datasets: OSDD (Gouidis et al. 2022), CGQA-States (Naeem et al. 2021), MIT-States (Isola, Lim, and Adelson 2015), VAW (Pham et al. 2021). As a frame of reference, the performance of a supervised visual state classification model is included. This reference model relies on the ResNet-101 network architecture, trained in a fully supervised setting on each dataset independently.

compassing object state classes as a distinct subset within its attribute annotations. Similarly, MIT-States (Isola, Lim, and Adelson 2015) and CGQA-States (Mancini et al. 2022) are widely employed datasets within the domain of attribute classification.

Metrics: Our evaluation protocol adheres to the established zero-shot evaluation method as outlined in (Purushwalkam et al. 2019). In contrast to the conventional setting where accuracy across all classes is typically reported, our approach involves computing accuracy for each class individually. Subsequently, an overall mean average is calculated across these individual class results. This method assigns equal weight to each class, irrespective of the respective number of samples associated with each class.

Ablation Study

The ablation study focuses on addressing the following key questions:

- Can domain-specific knowledge generated by an LLM contribute to resolving the zero-shot problem under examination?

- What is the optimal representation method for domain-specific knowledge generated by an LLM?

- Can domain-specific knowledge from the LLM be effectively combined with general-purpose knowledge de-

rived from large corpora, and what is the optimal approach for this integration?

- To what extent does the size of the data used for obtaining general-purpose knowledge impact the performance of zero-shot methods?

Tables 3- 7 present the results of the ablation study with the reported scores in each table corresponding to the mean values and standard deviations of the performance across the 30 different KGs. In more detail, Table 3 shows results related to the corpus size used as input for the LLM. Table 4 contains the results for the impact of the representation used for the semantic vectors. Table 5 presents results concerning different combinations of domain-specific and general-purpose knowledge. Lastly, Tables 6 and 7 pertain to experiments involving the dimensionality reduction of the semantic vectors and the CNN model used as a classifier, respectively.

Impact of corpus size: Analyzing the impact of corpus size (Table 3), it is observed that the best results are achieved with the basic-sized corpus rather than the extended one. This can be attributed to the domain specificity of prompts used for the creation of the basic corpus since the prompts used for extending the corpus are more general. Additionally, in line with expectations from the literature (Lison and Kutuzov 2017; Rahimi and Homayounpour 2022) that shows that stopword removal is not beneficial in all down-

stream tasks, excluding stopwords from the basic corpus results in a decrease in performance.

Types of semantic embeddings: Analyzing the results related to different types of semantic embeddings (Table 4), the following observations can be made. For general-purpose embeddings, the best scores are achieved by GloVe 6B and fastText 16B, which have been trained on fewer tokens. This outcome supports the notion that embeddings from large corpora may be tainted with noise, thereby reducing performance. Regarding domain-specific embeddings generated by the LLM corpus, the best representation is GloVe, followed by fastText. Moreover, it is observed that domain-specific embeddings achieve performance comparable to that of general-purpose embeddings across all representation classes.

Fusion of domain-specific / general-purpose embeddings: The key findings from the fusion of domain-specific and general-purpose embeddings presented in Table 5 are as follows. The concatenation of the two types of embeddings leads to performance gains in most cases for GloVe 6B (except for the CGQA-States dataset) and in all cases for GloVe 42B. These gains can be attributed to domain-specific knowledge: fusion with random embeddings results in a significant decrease in performance. Moreover, the version of GloVe 6B embeddings that has the same coverage as the embedding of the basic corpus (GloVe 6B - BDCVR in the Table) scores lower than GloVe 6B. This result indicates that the combination of domain-specific and general-purpose embeddings achieves better performance *despite* their lower coverage.

Impact of dimensionality reduction: Analyzing the results for dimensionality reduction and fusion techniques in Table 6, it is observed that averaging is the least successful technique, achieving the worst results in the majority of cases. In the fusion of GloVe 6B with the basic corpus, the best scores are obtained with PCA, whereas Autoencoder achieves the best result in the combination of GloVe 6B with the extended corpus. Finally, results regarding the CNN model used as the State Classifier (Table 7) suggest that ResNet-101 outperforms ViT-16/B in most cases.

Experimental Results

To the best of our knowledge, aside OaSc (Gouidis et al. 2024), there currently is no readily available object-agnostic state model suitable for deployment in a zero-shot setting. Therefore, in addition to OaSc, we have chosen to employ additionally three state-of-the-art Vision Language Models (VLMs) that support this functionality: CLIP (Radford et al. 2021), ALIGN (Jia et al. 2021), and BLIP (Li et al. 2022). In our experimentation, we utilize three variants of CLIP one of ALIGN and one of BLIP. It is essential to note that all these models indirectly violate the fundamental assumptions of the zero-shot setting, as the training data includes pairs of text and images containing the target classes relevant to our task.

Table 8 presents the results of the experimental evaluation across the four datasets. The reported scores for our method correspond to the performance of the best ablated model,

which utilized the fusion of LLM-based embeddings from the extended corpus with GloVe 6B pre-trained embeddings. This model employed the Autoencoder as a dimensionality reduction technique and ResNet-101 as the classifier. Regarding the VLMs, following the typical procedure (Xue et al. 2022), we used 16 different prompts[1] and report the mean value of these scores.

The results indicate that our model not only outperforms OaSc by a significant margin but also achieves superior performance compared to every competing VLM across all datasets. These findings strongly support the robustness of our approach, especially considering that the competing VLMs are recognized state-of-the-art models for various zero-shot tasks.

Conclusions and Future Work

This work represents a step towards integrating an LLM into a hybrid framework for addressing a crucial Computer Vision problem. The experimental results strongly suggest that incorporating knowledge from the LLM can substantially enhance the performance of the framework. The success of this integration instills confidence in further exploration of ideas for leveraging LLMs in the near future.

Firstly, we consider that the utilization of the LLMs can be extended to the construction and refinement of KGs used for generating the visual embeddings essential for the zero-shot classification task. Specifically, by employing tailored prompts, the LLM can generate text, which, after parsing, will yield a KG. Similarly, querying the LLM about an existing KG could facilitate the addition or pruning of edges. Furthermore, the LLM can play a role in constructing the KG used for training the GNN which is essential for the projection of the semantic embeddings to the visual space.

In this work we utilize a 13B parameter model to strike a reasonable balance between cost and performance, The obtained results are highly promising as they already advance the state-of-the-art performance on the problem. However, employing a more robust model might enhance the capture of commonsense knowledge and human reasoning capabilities. Additionally, searching for the optimal set of prompts remains an open subject, as it plays a crucial role in determining the quality of the LLM-based embeddings. Finally, we aim to further enhance the capabilities of the LLM model by fine-tuning it using publicly available image-text datasets, such as Visual Genome.

Acknowledgements

References

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5: 135–146.

Coates, J.; and Bollegala, D. 2018. Frustratingly Easy Meta-Embedding – Computing Meta-Embeddings by Averaging Source Word Embeddings. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 194–198. New Orleans, Louisiana: Association for Computational Linguistics.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.

Faruqui, M.; Dodge, J.; Jauhar, S. K.; Dyer, C.; Hovy, E. H.; and Smith, N. A. 2014. Retrofitting Word Vectors to Semantic Lexicons. *CoRR*, abs/1411.4166.

Fukumizu, K.; Bach, F. R.; and Gretton, A. 2007. Statistical Consistency of Kernel Canonical Correlation Analysis. *J. Mach. Learn. Res.*, 8: 361–383.

Ghannay, S.; Favre, B.; Estève, Y.; and Camelin, N. 2016. Word Embedding Evaluation and Combination. In Calzolari, N.; Choukri, K.; Declerck, T.; Goggi, S.; Grobelnik, M.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 300–305. Portorož, Slovenia: European Language Resources Association (ELRA).

Gong, J.; Liu, W.; Pei, M.; Wu, C.; and Guo, L. 2022. ResNet10: A lightweight residual network for remote sensing image classification. In *2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 975–978.

Gouidis, F.; Papoutsakis, K.; Patkos, T.; Argyros, A.; and Plexousakis, D. 2024. Exploring the Impact of Knowledge Graphs on Zero-Shot Visual Object State Classification. In *Proceedings of the the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*.

Gouidis, F.; Patkos, T.; Argyros, A.; and Plexousakis, D. 2022. Detecting Object States vs Detecting Objects: A New Dataset and a Quantitative Experimental Study. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, volume 5, 590–600.

Huang, N.; Deshpande, Y. R.; Liu, Y.; Alberts, H.; Cho, K.; Vania, C.; and Calixto, I. 2022. Endowing Language Models with Multimodal Knowledge Graph Representations. arXiv:2206.13163.

Isola, P.; Lim, J. J.; and Adelson, E. H. 2015. Discovering states and transformations in image collections. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June: 1383–1391.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, 4904–4916. PMLR.

Kampffmeyer, M.; Chen, Y.; Liang, X.; Wang, H.; Zhang, Y.; and Xing, E. P. 2019. Rethinking knowledge graph propagation for zero-shot learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June: 11479–11488.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; Bernstein, M. S.; and Fei-Fei, L. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vision*, 123(1): 32–73.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Lai, S.; Liu, K.; He, S.; and Zhao, J. 2016. How to Generate a Good Word Embedding. *IEEE Intelligent Systems*, 31(6): 5–14.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 12888–12900. PMLR.

Lison, P.; and Kutuzov, A. 2017. Redefining Context Windows for Word Embedding Models: An Experimental Study. In Tiedemann, J.; and Tahmasebi, N., eds., *Proceedings of the 21st Nordic Conference on Computational Linguistics*, 284–288. Gothenburg, Sweden: Association for Computational Linguistics.

Mancini, M.; Naeem, M. F.; Xian, Y.; and Akata, Z. 2022. Learning Graph Embeddings for Open World Compositional Zero-Shot Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8828(c): 1–15.

Massarenti, N.; and Lazzarinetti, G. 2021. Dialog Clustering: a Framework for Automatic Text Clustering. In *AIABI@AI*IA*.

McInnes, L.; Healy, J.; Saul, N.; and Großberger, L. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29): 861.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In Burges, C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Miller, G. A. 1994. WordNet: A Lexical Database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Naeem, M.; Xian, Y.; Tombari, F.; and Akata, Z. 2021. Learning Graph Embeddings for Compositional Zero-shot Learning. In *34th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Nayak, N. V.; Yu, P.; and Bach, S. H. 2022. Learning to compose soft prompts for compositional zero-shot learning. *arXiv preprint arXiv:2204.03574*.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2023. Unifying Large Language Models and Knowledge Graphs: A Roadmap. arXiv:2306.08302.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Pham, K.; Kafle, K.; Lin, Z.; Ding, Z.; Cohen, S.; Tran, Q.; and Shrivastava, A. 2021. Learning To Predict Visual Attributes in the Wild. In *Proceedings of the IEEE/CVF CVPR*, 13018–13028.

Pratt, S.; Liu, R.; and Farhadi, A. 2022. What does a platypus look like? Generating customized prompts for zero-shot image classification. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 15645–15655.

Purushwalkam, S.; Nickel, M.; Gupta, A.; and Ranzato, M. 2019. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3593–3602.

Qi, P.; Dozat, T.; Zhang, Y.; and Manning, C. D. 2018. Universal Dependency Parsing from Scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 160–170. Brussels, Belgium: Association for Computational Linguistics.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rahimi, Z.; and Homayounpour, M. M. 2022. The impact of preprocessing on word embedding quality: a comparative study. *Lang. Resour. Eval.*, 57(1): 257–291.

Rettig, L.; Audiffren, J.; and Cudre-Mauroux, P. 2019. Fusing Vector Space Models for Domain-Specific Applications. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 1110–1117. Los Alamitos, CA, USA: IEEE Computer Society.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Sap, M.; Le Bras, R.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N. A.; and Choi, Y. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press. ISBN 978-1-57735-809-1.

Sarma, P. K.; Liang, Y.; and Sethares, W. 2018. Domain Adapted Word Embeddings for Improved Sentiment Classification. In Haffari, R.; Cherry, C.; Foster, G.; Khadivi, S.; and Salehi, B., eds., *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, 51–59. Melbourne: Association for Computational Linguistics.

Sartori, G.; and Orrù, G. 2023. Language models and psychological sciences. *Frontiers in Psychology*, 14.

Seyler, D.; and Zhai, C. 2020. A Study of Methods for the Generation of Domain-Aware Word Embeddings. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 1609–1612. New York, NY, USA: Association for Computing Machinery. ISBN 9781450380164.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 4444–4451. AAAI Press.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; and Almahairi, A. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, 1096–1103. New York, NY, USA: Association for Computing Machinery. ISBN 9781605582054.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.

Wang, X.; Gao, T.; Zhu, Z.; Zhang, Z.; Liu, Z.; Li, J.; and Tang, J. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics*, 9: 176–194.

Wang, X.; Ye, Y.; and Gupta, A. 2018. Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6857–6866.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2018a. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9): 2251–2265.

Xian, Y.; Lorenz, T.; Schiele, B.; and Akata, Z. 2018b. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5542–5551.

Xue, H.; Sun, Y.; Liu, B.; Fu, J.; Song, R.; Li, H.; and Luo, J. 2022. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv preprint arXiv:2209.06430*.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1(January): 57–81.