On The Importance of Symbol Grounding and Top-Down Processes in Computer Vision

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht von
Sahand Sharifzadehgolpayegani
am 21.11.2022

1. Gutachter:	Prof. Dr. Volker Tresp
2. Gutachter:	Prof. Dr. Nassir Navab
3. Gutachter:	Prof. Dr. Vasileios Belagiannis
Tag der mündlichen Prüfung:	07.02.2023

Eidesstattliche Versicherung

Hiermit erkläre ich, Sahand Sharifzadehgolpayegani, an Eides statt, dass die vorliegende Dissertation ohne unerlaubte Hilfe gemäß Promotionsordnung vom 12.07.2011, §8, Abs. 2 Pkt. 5, angefertigt worden ist.

München, 21.11.2022 _____

Sahand Sharifzadehgolpayegani

Contents

List of Publications

Main Publications*

- **Sharifzadeh, S.**, Baharlou, S. M., Schmitt, M., Schütze, H., & Tresp, V. (2022, May). Improving Scene Graph Classification by Exploiting Knowledge from Texts. *In Proceedings of the AAAI Conference on Artificial Intelligence.*

  9 pages, double column, https://doi.org/10.48550/arXiv.2203.10202

- **Sharifzadeh, S.**, Baharlou, S. M., & Tresp, V. (2021, May). Classification by Attention: Scene Graph Classification with Prior Knowledge. *In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 6, pp. 5025-5033).*

  9 pages, double column, https://doi.org/10.48550/arXiv.2011.10084

- **Sharifzadeh, S.**, Baharlou, S. M., Berrendorf, M., Koner, R., & Tresp, V. (2021, January). *Improving visual relation detection using depth maps. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 3597-3604). IEEE.*

  8 pages, double column, http://dx.doi.org/10.1109/ICPR48806.2021
  .9412945

- Tresp, V., **Sharifzadeh, S.**, & Konopatzki, D. (2019). A model for perception and memory. *In Conference on Cognitive Computational Neuroscience.*

  4 pages, double column, http://dx.doi.org/10.32470/CCN.2019.1264-0

- Schmitt, M., **Sharifzadeh, S.**, Tresp, V., & Schütze, H. (2020, November). An Unsupervised Joint System for Text Generation from Knowledge Graphs and Semantic Parsing. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 7117-7130).*

  13 pages (14), double column, http://dx.doi.org/10.18653/v1/2020.emnlp-main.577

Other Publications

- **Sharifzadeh, S.**, Chiotellis, I., Triebel, R., & Cremers, D. (2016). Learning to drive using inverse reinforcement learning and deep q-networks. *Advances in Neural Information Processing Systems 29 (NeurIPS 2016), Workshop on Deep Learning for Action and Interaction.*

  7 pages. https://doi.org/10.48550/arXiv.1612.03653

- Tresp, V., **Sharifzadeh, S.**, Li, H., Konopatzki, D., & Ma, Y. (2021). The Tensor Brain: A Unified Theory of Perception, Memory and Semantic Decoding. *Under Review.*

  38 pages, https://doi.org/10.48550/arXiv.2001.11027

- Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., ..., **Sharifzadeh, S.**, ... & Simonyan, K. (2022). Flamingo: a visual language model for few-shot learning. arXiv preprint arXiv:2204.14198.

  66 pages. https://doi.org/10.48550/arXiv.2204.14198

- Carrami, E. M., **Sharifzadeh, S.**, Wietek, N. C., Artibani, M., El-Sahhar, S., Sauka-Spengler, T., ... & Ahmed, A. A. (2020). A highly accurate platform for clone-specific mutation discovery enables the study of active mutational processes. *Elife, 9, e55207.*

  21 pages, https://doi.org/10.7554/eLife.55207

- Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., Li, H., Pan, J., **Sharifzadeh, S.**, ... & Menze, B. (2022). Relationformer: A Unified Framework for Image-to-Graph Generation. *European Conference on Computer Vision (ECCV).*

  14 pages.

- Babaians, E., Sharma, T., Karimi, M., **Sharifzadeh, S.**, & Steinbach, E. (2022). PourNet: Robust Robotic Pouring Through Curriculum and Curiosity-based Reinforcement Learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

  8 pages, double column.

- Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Galkin, M., **Sharifzadeh, S.**, ... & Lehmann, J. (2021). Bringing light into the dark: A large-scale evaluation of

knowledge graph embedding models under a unified framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

20 pages, double column, http://dx.doi.org/10.1109/TPAMI.2021.3124805

- Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., **Sharifzadeh, S.**, Tresp, V., & Lehmann, J. (2021). PyKEEN 1.0: a Python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research, 22(82), 1-6.*

  6 pages. https://doi.org/10.48550/arXiv.2007.14175

- Hashemi, H., Hartmann, N., **Sharifzadeh, S.**, Srebre, M., & Kuhr, T. (2022). IEA-GAN: Intra-Event Aware GAN for the Fast Simulation of PXD Background at Belle II. *Inter-experiment Machine Learning Workshop (CERN).*

  24 pages.

- Kougia, V., Baharlou, S., **Sharifzadeh, S.**, & Roth, B. (2022). MemeGraphs: Linking Memes to Knowledge Graphs. *Under Review.*

  9 pages, double column.

* In all my publications I have used a shorter version of my family name, *Sharifzadeh*, instead of *Sharifzadehgolpayegani*.

Abstract

In the past decade, feedforward artificial neural networks have stormed the field of artificial intelligence and shown impressive results in many domains. Nevertheless, one of the challenges in artificial intelligence is connecting the differentiable feature space in deep learning to the rich world of object-based, symbolic knowledge. For example, in computer vision, images consist of different features, such as edges and curves at a lower level, while at a higher level, they include objects and relations. Even though it is not feasible to describe the low-level features using the natural language, the attributes and relations between objects can be represented by symbols and are well-documented throughout human literature. Therefore, developing novel and effective architectures that can learn and utilize symbolic knowledge within the differentiable deep learning framework is essential. To this end, in this dissertation we argue for methods that map symbols to image-grounded representations such that they share the same representation space as images.

Furthermore, we discuss the key role of top-down processes in utilizing object-level knowledge; top-down signals have been shown to play a significant role in the human brain for overcoming challenges such as occlusion. For example, even though there might not be enough pixels from a truck's wheel in an image, after detecting the truck itself within the top layers of a neural network, we can use the higher-level knowledge to recognize a small area in a corner that corresponds to the wheel. Nevertheless, current feedforward neural networks lack effective inductive biases for top-down processing. We show that grounding symbols in images and employing top-down mechanisms not only improves the scene understanding but also allows us to benefit from the massive pool of human-written symbolic knowledge in addition to image annotations.

In summary, this dissertation introduces significant advances in the artificial intelligence domain, particularly computer vision and modeling commonsense. We propose models that utilize (1) structured knowledge, (2) unstructured text, and (3) 3d information to improve scene understanding, and through large-scale experiments, we show that our models significantly improve state-of-the-art results.

Zusammenfassung

In den letzten zehn Jahren haben künstliche neuronale Feedforward-Netzwerke das Feld der künstlichen Intelligenz gestürmt und in vielen Bereichen beeindruckende Ergebnisse gezeigt. Dennoch besteht eine der Herausforderungen im Bereich der künstlichen Intelligenz darin, den differenzierbaren Merkmalsraum des Deep Learning mit der reichen Welt des objektbasierten, symbolischen Wissens zu verbinden. In der Computer Vision beispielsweise bestehen Bilder aus verschiedenen Merkmalen, wie Kanten und Kurven auf einer niedrigeren Ebene, während sie auf einer höheren Ebene Objekte und Beziehungen enthalten. Auch wenn es nicht möglich ist, die Merkmale der unteren Ebene mit Hilfe der natürlichen Sprache zu beschreiben, können die Attribute und Beziehungen zwischen Objekten durch Symbole dargestellt werden und sind in der menschlichen Literatur gut dokumentiert. Daher ist die Entwicklung neuartiger und effektiver Architekturen, die symbolisches Wissen im Rahmen des differenzierbaren Deep Learning erlernen und nutzen können, von entscheidender Bedeutung. Zu diesem Zweck plädieren wir für Methoden, die Symbole auf bildbasierte Repräsentationen abbilden, so dass sie denselben Repräsentationsraum wie Bilder teilen.

Darüber hinaus erörtern wir die Schlüsselrolle von Top-Down-Prozessen bei der Nutzung von Wissen auf Objektebene; es hat sich gezeigt, dass Top-Down-Signale im menschlichen Gehirn eine wichtige Rolle bei der Bewältigung von Herausforderungen wie Verdeckungen spielen. Auch wenn beispielsweise nicht genügend Pixel eines LKW-Rads in einem Bild vorhanden sind, können wir nach der Erkennung des LKWs in den obersten Schichten eines neuronalen Netzwerks das Wissen auf höherer Ebene nutzen, um einen kleinen Bereich in der Ecke zu erkennen, der dem Rad entspricht. Allerdings fehlt es den aktuellen neuronalen Feedforward-Netzen an effektiven induktiven Vorspannungen für die Top-down-Verarbeitung. Wir zeigen, dass die Verankerung von Symbolen in Bildern und der Einsatz von Top-Down-Mechanismen nicht nur das Verständnis der Szene verbessert, sondern es uns auch ermöglicht, zusätzlich zu den Bildannotationen von dem riesigen Fundus an von Menschen geschriebenem symbolischem Wissen zu profitieren.

Zusammenfassend lässt sich sagen, dass diese Dissertation bedeutende Fortschritte im

Bereich der künstlichen Intelligenz, insbesondere im Bereich des Computersehens und der Modellierung des gesunden Menschenverstandes, vorstellt. Wir schlagen Modelle vor, die (1) strukturiertes Wissen, (2) unstrukturierten Text und (3) 3D-Informationen nutzen, um das Verstehen von Szenen zu verbessern, und durch groß angelegte Experimente zeigen wir, dass unsere Modelle den Stand der Technik deutlich verbessern.

List of Publications and Declaration of Authorship

- Sharifzadeh et al. [2021]

  The research idea was developed and conceptualized by Sahand Sharifzadeh. Sahand Sharifzadeh did the main part of the model implementation and design of the experiments. Sina Baharlou contributed to parts of the pre-processing and evaluation code, and taking care of the data splits. The final manuscript was mainly written by Sahand Sharifzadeh and revised by all authors.

  This publication serves as Chapter 2 of this thesis.

- Sharifzadeh et al. [2022]

  The research idea was developed and conceptualized by Sahand Sharifzadeh. Sahand Sharifzadeh and Sina Baharlou did the main part of the model implementation and design of the experiments and extracted the data splits. Martin Schmitt implemented and evaluated the text-to-graph model. The final manuscript was mainly written by Sahand Sharifzadeh and revised by all authors.

  This publication serves as Chapter 3 of this thesis.

- Sharifzadeh et al. [2020]

  The research idea was proposed by Sahand Sharifzadeh and discussed with Max Berrendorf. Sahand Sharifzadeh did the main implementation and conducted experiments; Sina Moayed Baharlou contributed to parts of the code, Max Berrendorf and Rajat Koner contributed smaller parts of the code. The final manuscript was mainly written by Sahand Sharifzadeh and revised by all authors.

  This publication serves as Chapter 4 of this thesis.

- Tresp et al. [2019]

  The research idea was developed by Volker Tresp and further developed by Volker Tresp and Sahand Sharifzadeh and discussed with Dario Konopatzki. Volker Tresp and Sahand Sharifzadeh designed the experiments. Sahand Sharifzadeh implemented the code for all experiments. The manuscript was mainly written by Volker Tresp and Sahand Sharifzadeh wrote the section on experiments. The final manuscript was revised by all authors.

  This publication serves as Chapter 5 of this thesis.

- Schmitt et al. [2019]

  The initial idea was brainstormed by Sahand Sharifzadeh and Martin Schmitt. The idea was further developed by Martin Schmitt. Martin Schmitt performed the implementations and evaluations. The final manuscript was mainly written by Martin Schmitt and revised by all authors.

  This publication serves as Chapter 6 of this thesis.

1 Introduction

1.1 Prologue

The reality we can put into words is never reality itself.

Werner Heisenberg

Words only give us descriptions, or "models" of the world, as Heisenberg points out. Models quench our thirst for understanding and, perhaps more importantly, help us predict the unknown. We first tried to model our world by describing it with simple words or "symbols". Our symbols evolved as the phenomena we sought to explain became more complex and high-dimensional. We went further away from our natural language and used mathematical symbols and equations to write and reason more efficiently. We modeled gravity, electromagnetism, and nuclear forces. We modeled chemistry, society, and economy. The further we went, the harder it became to find the relations between symbols in our models. When describing a simple daily phenomenon such as "what a cat looks like," we could no longer put the symbols together ourselves. When explaining the underlying behavior of cancer cells, our symbols fell short. We needed something that could describe indescribable, non-linear, and complex phenomena. Thus, we created Machine Learning.

Machine Learning has shown promising results in modeling complex and high-dimensional data. In particular, Deep Learning [LeCun et al., 2015] has fueled an acceleration in Artificial Intelligence research in the past decade. In computer vision, deep convolutional neural networks (CNNs), long-short term memories [Hochreiter and Schmidhuber, 1997], and Transformers [Vaswani et al., 2017] often surpass not only classical approaches in image understanding but also human performance. Artificial neural networks still use symbols and rules, i.e., vectors and operations, but their power is in their scalability, continuity, and differentiability; layers of matrices can adapt freely based on the gradient signals they receive from higher layers. We only need to define an objective and feed the

input. The difference between the predictions of the artificial neural networks and the targets gives us a signal to update our layers of matrices. As a result, we managed to build predictive models without manually dictating the symbolic rules [1].

Relying on gradient descent to find the model rules comes with costs; for example, we overlook that high-level symbol representations, although hidden, still emerge within layers of an artificial neural network. Acknowledging this fact can help us interpret the internal mechanisms of the learned models, design more effective neural network architectures (e.g., networks that are encouraged to create disentangled representations [Higgins et al., 2016] or image-grounded symbol representations [Sharifzadeh et al., 2021]), and bridge the gap between the two worlds of differentiable matrices in deep learning frameworks and the human written symbols [Sharifzadeh et al., 2022]. For example, we have created a massive library of knowledge over the past centuries. Scientists and engineers have documented the results of their costly, time-consuming, and sometimes unrepeatable experiments by putting them into words. We can benefit significantly by bridging the gap between human-written symbols and learnable matrices.

In this dissertation, we focus on two main concepts to help us design more effective architectures that acknowledge the presence of symbol representations and close the gap to symbolic knowledge. First is the role of top-down connections in the network, and second is the importance of grounding symbols in images. In particular, we study the role of these two concepts in classifying objects and their relations in images, a computer vision task known as scene graph classification. In this setting, we argue for architectures consisting of two major components: a *feature extraction backbone*, often modeled by a CNN that takes images as inputs and gives out object-based representations, and a *relational reasoning module*, often modeled with message propagation functions such as Graph Neural Networks that work with object-based representations as input. While modern representation learning techniques such as self-supervised methods [Chen et al., 2020a,b, Grill et al., 2020, Chen and He, 2021] allow us to train effective extractors of low-level image features without the need for human labels, there is a lot that we can learn about higher-level object relations by reading literature. As will be discussed, the relational reasoning module can work with different forms of object representations,

[1] In the classic AI, symbols mainly refer to high-level, human-comprehensible signs or words. However, symbols are any mark, sign, or word that represents an idea, object, or relationship. This includes *any* words or even math symbols. Therefore, here we use the term "symbol" in its general form to discuss how humans started building models by putting together the symbols, and now deep learning continues to do the same. Later, when we talk about the symbol grounding problem, we use the term "symbol" in its classic AI sense. We clarify this using adjectives such as "high-level" or "human-readable".

including image-grounded symbol representations. In what follows, we briefly introduce each of the mentioned concepts.

1.1.1 Top-down Connections

Feedforward artificial neural networks often fail under challenging scenarios in computer vision, such as occlusions or variations in lighting conditions, viewpoints, and object pose. Solving challenges such as occlusion seem to involve recurrent processes in the human brain [Johnson and Olshausen, 2005, Wyatte et al., 2014, Tang et al., 2014]; in contrast to computer vision models, where the inputs are an instant blink of pixels with no context, for humans, the visual experience is a continuous flow of sensory inputs where inferences on previous inputs recurrently affect the inferences on upcoming ones. In fact, according to anatomical and functional data, object recognition in humans is significantly influenced by feedback connections [Spoerer et al., 2017], and feedback connections in the ventral visual pathway have been shown to have similar densities as feedforward connections [Felleman and Van Essen, 1991, Sporns and Zwi, 2004, Markov et al., 2014]. From a cognitive perspective on an object level, recurrent connections can, for example, help us make better sense of the scene: as we look at an object in a scene, our working memory is already filled with the impression of neighboring objects that we saw a few moments ago, and through feedback processes, it is loaded with recollections of relevant past experiences and knowledge [Tresp et al., 2019, 2020]. However, the most popular architectures for modeling computer vision tasks are feedforward neural networks that either lack recurrent processing completely or apply it only in one direction where the future predictions never influence the predictions in the past.

This thesis argues for novel inductive biases required to implement feedback processes within artificial neural network-based frameworks. In particular, we model top-down and lateral connectivities using graph transformers and attention functions [Sharifzadeh et al., 2021]. From a computational perspective, the lateral connections between several objects in a scene, e.g., using Graph Convolutional Neural Networks [Kipf and Welling, 2016] or LSTMs [Hochreiter and Schmidhuber, 1997], can contextualize the signals while top-down connections allow us to exploit higher-level information. For example, let us consider an image of a bowl full of fruits where the bowl has very few visible pixels, and therefore, a typical feedforward model fails to classify it. However, once we detect the neighboring objects such as fruits, our prior knowledge can tell us that "fruits typically go in bowls." The prior knowledge might also tell us that "fruits typically grow on trees." To use this

top-down information and infer correctly, we must recurrently combine the bottom-up visual cues with top-down signals while propagating these signals laterally between all the neighboring objects.

We evaluate our models and test our hypotheses by running large-scale experiments on Visual Genome [Krishna et al., 2017], a dataset with thousands of images annotated with their scene graphs (a set of triples that describe the objects and their relations in images). We show that our proposed architecture can accurately learn commonsense relational knowledge and that the top-down injection of this knowledge to scene representations leads to significantly higher classification performance [Sharifzadeh et al., 2021].

Other than exploiting top-down and object-level *relational* information in [Sharifzadeh et al., 2021], we study the effect of object-level *3D-information* on visual relation detection in Sharifzadeh et al. [2020]. To this end, we use a pre-trained network [Laina et al., 2016] capable of generating depth maps from RGB images as our model of "3D commonsense". Again, through bottom-up, top-down, and lateral connections, we infer and propagate the 3D information throughout the image and demonstrate its essential role in scene graph classification. Additionally, we release a new dataset of synthetically generated depth maps from Visual Genome [Sharifzadeh et al., 2020].

1.1.2 Symbol Grounding

One of the characteristics of human intelligence is that we can improve our scene understanding through reading books or communicating. This is related to the problem of symbol grounding, which explores the question "*Where do symbols get their meanings?*" [Harnad, 1990]. Mental representation for objects and concepts is widely discussed in epistemology [Pitt, 2020], and there are different opinions on how these representations come to be [Kant, 1787]. For example, what do we imagine when thinking of an entity such as a "*bird*" or an action such as "*two people playing basketball*"? In the Theory of Forms, Plato proposes "Forms" as the unchanging and absolute representation of each entity from the realm of forms that are independent of our experiences in this world. On the other hand, in the theory of cognitive development, Piaget suggests that we acquire knowledge representations from our prior observations and calls them *schemata* [Piaget, 1923]. According to Piaget, when an object is being perceived, the mind assigns it to a schema (a process called *assimilation*). By relational reasoning over schemata, assimilation helps to predict the facts surrounding the observation [Arbib, 1992]. If an observation contradicts prior schemata, we alter our schemata to *accommodate* the new fact. Similar

ideas have also been discussed in other fields, such as *embodied representations* in cognitive linguistics [Evans, 2006]. Grounding symbols in perceptions is one of the main focuses of this thesis. We borrow the terms schemata, assimilation, and accommodation from Piaget and often discuss them in the following chapters.

From a computational perspective, grounding symbol representations into perceptions leads to better model generalization [Sharifzadeh et al., 2021]. For example, as soon as we see an imaginary animal, such as a Bantha, we guess that we can ride it, even if we have never heard of or read about Banthas. This is because we can generalize from our prior perceptual knowledge of similar animals, such as horses.

Other than more generalization, when grounding meets top-down processing, it can enable multi-modal knowledge acquisition [Sharifzadeh et al., 2020, 2021, 2022]. For example, consider teaching a kid about zebras by describing them as striped horses [Harnad, 1990]; when we read or hear about novel facts, we can combine our existing symbol representations to compose an embedding for the new fact. Even though we have never perceived this new fact and cannot ground it in visual observation, we can create a synthetic grounded representation for this symbol by composing it from previously grounded representations. In fact, a study by St-Louis et al. [2008] showed that when the participants in a *visual* pattern recognition study are told the pattern's rules *verbally*, not only do they learn to categorize the patterns correctly, but their perception seems to change such that they can see the members and the non-members of the category as looking more different.

Computationally, this form of learning can be initiated by a top-down process since it activates the representations through the communicated symbolic knowledge rather than the bottom-up visual inputs. The perception then needs to adapt to accommodate this new visual rule. Therefore, the new representation's top-down signal has to go through the path of a bottom-up signal such that we can adapt and fine-tune our perceptual understanding as if the knowledge came from an actual sensory experience. In other words, top-down processing and grounding go hand in hand, which is the reason to study them both together in this dissertation.

Our experiments on the Visual Genome dataset show that with this form of learning, we can learn to recognize novel objects and relations without using annotated training images and, instead, using curated facts in the form of Knowledge Graphs [Sharifzadeh et al., 2021] or using textual descriptions [Sharifzadeh et al., 2022]. In fact, we can achieve similar accuracy in relation detection with only 1% of the annotated images and instead use textual descriptions.

1.2 Overview of Contributions

Here we give an overview of the thesis and the positions of the included publications. We organize the research works in order of importance rather than the chronological order.

- Section 1.3 gives an overview of the broader research area. It reviews and orders existing work and introduces concepts and formalisms commonly employed in the subsequent chapters. It provides a detailed overview of representation learning, scene graphs models, models of commonsense including knowledge graphs and language models, visual language models, and the symbol grounding problem.
- Chapter 2 addresses top-down processes and their connection to symbol grounding. In particular, in this chapter, we introduce *schemata*, a form of image-grounded vector representations that model the visual-relational commonsense of each class. We define classification as an attention layer between the bottom-up image-based representations and the symbol-based representations (schemata). We show that schemata learn to capture commonsense knowledge accurately and that the iterative, top-down injection of this knowledge to scene representations leads to significantly higher classification performance. Additionally, grounding schemata in images, and having a top-down approach to inject them into the classification pipeline, enables us to introduce a new learning mechanism. In this mechanism, instead of using annotated images, we can improve the classification pipeline using purely symbolic, relational data from knowledge graphs by employing their image-grounded symbol representations (schemata). Combined with a self-supervised backbone and with 1% of annotated images only, this gives more than 36% accuracy in predicate prediction, 3% in object classification, and 26% in scene graph classification accuracy.
- Chapter 3 introduces *texema*, an architecture for scene graph classification that can be fine-tuned from scene descriptions in natural language instead manually crafted knowledge graphs (such as those in the previous chapter) or annotated images. Texema relies on a transformer-based model that can convert the unstructured natural language into the structured form of knowledge graphs. The generated knowledge graphs can then be mapped to image-based representations using the pre-trained class prototypes (schemata), and treated as if they come from an actual image. We use these synthetic image-based representations to fine-tune the classification pipeline. We show that this process leads to 8x more accurate

results in scene graph classification, 3x in object classification, and 1.5x in predicate classification, compared to the supervised baselines with only 1% of the annotated images.

- Chapter 4 studies the effect of synthetically generated depth maps in visual relation detection. While most visual relation detection approaches rely on object information extracted from RGB images, such as 2D bounding boxes, feature maps, and predicted class probabilities, in this chapter, we argue for the importance of 3D information provided by depth maps. In order to obtain depth maps, one does not need to rely on specific equipment since they can be generated synthetically from RGB images in a bottom-up process and using a pre-trained convolutional neural network. The depth information can then be fused with the RGB image information before classifying the relations. To enable this study, we release a new dataset of synthetically generated depth maps, *VG-Depth*, as an extension to Visual Genome (VG). Additionally, we introduce a novel metric better suited for reflecting the experimental results, specially given the imbalanced distribution of VG annotations.

- Chapter 5 analyzes the close link between human perception and memory. In this chapter, we propose a biologically plausible, computational cognitive model to capture the interaction between episodic memory, semantic memory, and working memory. Our experiments on the Stanford Visual Relation Data (VRD) demonstrate that semantic memory can evolve from perception as a distinguishable functional module. There is a close link between the cognitive model introduced in this chapter and the model in Chapter 2.

- Chapter 6 proposes the first unsupervised architecture for text-to-graph (semantic parsing) and graph-to-text (text generation). Unlike previous works, this model does not require parallel graph-text training data and does not need to rely on domain adaptation techniques to transfer well to different domains. We evaluate our approach on the WebNLG and a new benchmark we create from the scene graphs of Visual Genome. Our system outperforms strong baselines for both conversion tasks without manual adaptation from one dataset to the other. In additional experiments, we investigate the impact of using different unsupervised objectives. The findings of this chapter were influential in designing the supervised text to graph model that we introduced in Chapter 3.

In summary, we believe this work significantly advances the field of scene understanding,

particularly scene graph classification, where the results of our proposed models are among the state-of-the-art to this date. As a result of this work, we can now model commonsense directly from images instead of handcrafting it and take a step towards artificial general intelligence (AGI) by enabling embodied knowledge acquisition. Furthermore, we improve the perception models by introducing iterative feedback signals from the commonsense representations to the vision pipeline. Moreover, by lifting the burden of extensive labor for image annotation and using textual knowledge transfer, we can now fine-tune scene graph classification models in an economically and computationally more viable way. Finally, by creating an external dataset of synthetic depth maps, we show the importance of 3D information in scene understanding and pave the way for future research in multimodal scene graph classification.

As reproducible science is of the highest importance to us, we provide open-sourced implementations for most of the work presented and explicitly reference them in each chapter.

1.3 Background

This chapter introduces the central concepts of the thesis in greater detail than a single paper allows. It also reviews existing works to contextualize the thesis' contributions better. We begin by introducing general notation in Section 1.3.1. We then discuss Representation Learning and its variations, particularly recent advancements in self-supervised learning in Section 1.3.2. Scene Graphs are introduced in Section 1.3.3 including different detection approaches, evaluation tasks, metrics, and datasets. We then define commonsense in Section 1.3.4 and introduce structured and unstructured methods of modeling commonsense using knowledge graphs embeddings models (Section 1.3.4) and language models (Section 1.3.4). Section 1.3.5 briefly introduces the recent Visual Language Models. Finally, Section 1.3.6 discusses the symbol grounding problem and motivates the upcoming works in this dissertation.

1.3.1 Notation

In this section, we introduce the notation used throughout the remainder of this chapter. The notation introduced here is also mostly consistent with the individual publications in the other chapters. In general, we use lower-case Greek letters, e.g., α, to denote scalar values, lower-case bold-font letters, e.g., $\mathbf{x}$, to denote vectors, upper-case bold-font, e.g.,

$\mathbf{X}$, to denote matrices or higher-order tensors. We denote real and complex numbers by $\mathbb{R}$ by $\mathbb{C}$. If not noted otherwise, we assume each variable to be real. With log we refer to the natural logarithm (ln) if not stated otherwise. By $\langle \mathbf{x}, \mathbf{y} \rangle$ we denote the inner product between $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. If not noted otherwise, we use the standard inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$. By $\|\mathbf{x}\|$ we denote the norm of $\mathbf{x}$, and by $\|\mathbf{x}\|_p = \left(\sum_{i=1}^{d} |\mathbf{x}_i|^p\right)^{1/p}$ the p norm specifically.

We commonly use the following activation functions. When applied to vectors, matrices, or tensors, we realize them as elementwise operations.

- The Rectified Linear Unit (ReLU)

$$\mathrm{ReLU}(\alpha) = \max\{0, \alpha\}$$

- The Leaky ReLU [Maas et al., 2013]:

$$\mathrm{LeakyReLU}(\alpha) = \begin{cases} \alpha & \text{if } \alpha > 0 \\ \beta \cdot \alpha & \text{otherwise} \end{cases}$$

  where $\beta = 10^{-2}$ if not specified differently.

- The (logistic) sigmoid function

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)}$$

- The softmax

$$(\mathrm{softmax}(\mathbf{x}))_i = \frac{\exp(\mathbf{x}_i)}{\sum_j \exp(\mathbf{x}_j)}$$

  With slight abuse of notation, we use the same function symbol with an additional argument to denote the vectorized softmax operation applied to the rows/columns of a matrix. The second argument denotes the axis along which the normalization is applied, e.g., $\mathrm{softmax}(\mathbf{X}, 1)$ denotes the row-wise softmax.

1.3.2 Representation Learning

There are many ways to represent information. Depending on the representation of the information, some tasks can be simple or very difficult [Goodfellow et al., 2016]. For example, finding the direction from location A to location B is much easier for us, given

the city map rather than looking at a graph with nodes representing streets and edges representing their connection. One of the challenges in deep learning is learning effective representations from data such that we can solve the downstream tasks. In supervised learning, the tasks are given explicitly, and the representations are shaped to handle the supervised task. As a result, the representation highly depends on the annotated training data and may lack sufficient generalization that allows us to transfer them to new tasks. On the other hand, unsupervised or self-supervised learning approaches aim to extract representations without any labels. As a result, they generalize better to downstream tasks and allow us to utilize the massive set of available unlabeled data without the need for human labor. In humans, the sparse supervisory signals from the environment appear to have also led to the development of self-supervised learning methods in the brain. Since most of the work in this dissertation relies on self-supervised representation learning methods, in the following, we discuss some of the most recent advancements in self-supervised learning for images and also for symbolic data.

Self-supervised Learning for Images

Most self-supervised approaches rely either on a generative or a discriminative objective [Chen et al., 2020a]. In generative approaches, the goal is to learn representations that can reconstruct the inputs. This approach is widely used in modern language models where the inputs are sentences, and the targets are a masked version of the same input. Similarly, in computer vision, generative self-supervised methods, also known as unsupervised methods, try to reconstruct a given image input [Hinton et al., 2006, Kingma and Welling, 2013, Goodfellow et al., 2014]. However, as Chen et al. [2020a] argues, pixel-level generation is computationally expensive and may not be necessary for learning representations that can be used for other downstream tasks such as classification. Discriminative approaches on the other hand, rely on pretext tasks such that the learning objective is similar to those of supervised learning. Some of the most promising approaches in discriminative self-supervised learning are based on contrastive learning in the latent space [Hadsell et al., 2006, Dosovitskiy et al., 2014, Oord et al., 2018, Bachman et al., 2019, Chen et al., 2020a].

Recent methods in discriminative self-supervised learning achieve comparable or superior results to supervised learning on ImageNet [Deng et al., 2009]. For example, SimCLR [Chen et al., 2020a] and SimCLR-v2 [Chen et al., 2020b] augment the input images, feed them through the same convolutional neural network (a siamese neural

network architecture), and applies a contrastive loss such that the output of the original image, and its augmentation, are as similar as possible. In order to prevent the neural network to collapse (such that it learns to "cheat" by outputting a constant representation for all possible inputs), they use negative sampling; given two different augmentations **a** and **b** of an input image, and a different image **c**, the extracted representations from **a** compared to **b**, should be as similar as possible while the extracted representations from **a** and **b**, compared to **c** should be as dissimilar as possible. In an upcoming work, Grill et al. [2020] proposed an approach called BYOL (**B**ootstrap **Y**our **O**wn **L**atent) that does not require the expensive step required for sampling and training with negative data points. Instead, they used a momentum encoder. In a momentum encoder, the second positive sample (**b**) is fed into a frozen and offline version of the network that the first image (**a**) is fed into. The offline network is only updated every few steps such that it is always slightly different than the online network. Later, SimSiam [Chen and He, 2021] showed that BYOL works even without the momentum encoder, and the crucial step for BYOL is having the stop-gradients for the second network. VICReg [Bardes et al., 2021] challenged the previous works by showing that in order to prevent collapse, we only need two regularization terms: a term that maintains the variance of each embedding dimension above a threshold and a term that decorrelates each pair of variables. Other than the works that focus on approaches to prevent collapse, some focus on different aspects such as how to make more effective data augmentations or create better negative samples. In this dissertation, we often use self-supervised methods, in particular BYOL.

Self-supervised Learning for Symbols

In addition to the recent advancements in computer vision, self-supervised learning approaches are also often used for training language and knowledge graph models. In this scenario, the goal is to find representations for words (symbols). In Section 1.3.4, we will introduce knowledge graphs and languages as forms of modeling commonsense knowledge and briefly describe the self-supervised approaches in training knowledge graphs and language models.

1.3.3 Scene Graphs

A scene graph is a structured, symbolic description of an image [Krishna et al., 2017]. A scene graph can be represented as a set of triples defined as follows:

Definition 1.3.1 (Scene Graph)**.** *A Scene Graph (SG) is a 3-tuple* $\mathcal{SG} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ *with*

a set of entities $\mathcal{E}$, relations $\mathcal{R}$, and triples $\mathcal{T} \subseteq (\mathcal{E} \times \mathcal{R} \times \mathcal{E})$.

Each triple $(h, r, t) \in \mathcal{T}$ with *head* h, *relation* r, and *tail* entity t, represent a fact that is present in an image. For example, given an image of a man riding a horse on a grass field, we can define a scene graph as the set of triples (`Man`, `rides`, `Horse`) and (`Horse`, `on`, `Grass`). This representation is dual to a multi-relational graph view, where the heads and tails are nodes, and relations are directed edges from the head to the tail entity. In practice, when dealing with machine learning models for scene graph representation, instead of working directly with string values, e.g., *"Man"*, an integer-valued index is often assigned as the symbolic representation of the entities and relations, i.e., $\mathcal{E} = \{1, \ldots, |\mathcal{E}|\}$, or $\mathcal{R} = \{1, \ldots, |\mathcal{R}|\}$. This enables us to work with vectorized operations directly.

Scene Graph Detection

One of the fundamental tasks in Computer Vision is to take an image as input and generate its corresponding scene graph. This task is commonly known as *scene graph generation* or *scene graph detection (SGD)* [Krishna et al., 2017]. Extracting scene graphs from images can have many applications in downstream computer vision tasks such as image captioning, Visual Question Answering (VQA), image manipulation [Johnson et al., 2018], etc. Notably, it is possible to have end-to-end architectures that take images as input and solve the mentioned tasks. However, having an intermediate stage where we first extract symbolic scene graphs from an image before performing the downstream tasks has many advantages, including (a) contributing to the interpretability of the models, (b) creating a shared space where any data modalities such as images, sounds, or videos, can be mapped to, (c) creating a bridge to human language and the vast source of knowledge that it holds, and (d) creating a bridge to symbols and therefore, enabling logical reasoning operations, i.e., neuro-symbolic methods [Mao et al., 2019].

Approaches

The scene graph detection methods are often two-staged (e.g. Zellers et al. [2018], Lu et al. [2016], Sharifzadeh et al. [2021]), meaning that first a set of objects are detected from the image, and then the objects and their pairwise relations are classified, or sometimes single-staged [Liu et al., 2021, Cong et al., 2022, Shit et al., 2022], where the detection and classification are all together. The methods that we employ in this dissertation all have a two-staged approach.

Most state-of-the-art models for two-staged scene graph detection rely on the Faster R-CNN [Ren et al., 2015] framework that detects objects and extracts image-based object representations from the objects in each image. The Faster R-CNN is mainly equipped with a VGG-16 [Simonyan and Zisserman, 2014] or sometimes a ResNet-50 [He et al., 2016] backbone. It has been shown that using different forms of message passing between the extracted object representations in an image, e.g., RNNs [Xu et al., 2017], LSTMs [Zellers et al., 2018], Graph Convolutional Neural Networks [Yang et al., 2018], and Graph Transformers [Sharifzadeh et al., 2021], contributes to getting higher performance in classifying objects and relations.

Evaluation Tasks

In general, scene graph models are evaluated under three different settings: (1) scene graph detection (*SGDet*), where only the images are given, and the location, class, and size of objects are unknown, (2) scene graph classification (*SGCls*), where the location and size of objects in images are given as bounding boxes, and the goal is to find the suitable class for the objects and relations, (3) predicate classification (*PredCls*) where the location, size and class label of objects are given, and the goal is to detect and classify the relations (also known as predicates) between them.

Datasets

Stanford's Visual Relation [Lu et al., 2016] was one of the earliest image datasets with scene graphs annotations. Later, Visual Genome [Krishna et al., 2017] (VG) provided a larger dataset consisting of around 57k training images and their corresponding scene graphs as well as scene descriptions (in natural language). One of the most popular splits of Visual Genome that is also often used in this thesis is provided by Xu et al. [2017] and contains 150 of the most frequent objects in VG and 50 predicate classes, with an average of 11.5 objects and 6.2 predicates in each image.

Metrics

R@K The most commonly used metric for evaluating the experimental results of different scene graph classification models is Recall@K (R@K). In R@K, given each image and its predicted scene graphs, the predictions should first be sorted according to the prediction scores. Then, the ratio of ground truth labels that appear in the top-K scores are calculated. The final R@K measure is the mean accuracy among all images.

mR@K The distribution of labeled relations is often highly imbalanced. For example, in the Visual Genome test set, the predicate *wearing* appears 20,148 times, whereas the predicate *walking on* appears only 648 times [Sharifzadeh et al., 2020]. Therefore, if a model proposes a method to improve the prediction of *walking on*, the R@K cannot effectively reflect this improvement. In order to alleviate this problem, one can calculate the Macro Recall [Chen et al., 2019, Sharifzadeh et al., 2020] (mR@K). In this setting, the overall recall is computed by taking the mean over recall *per predicate*. Therefore, if the R@K of a specific predicate is largely improved, it will improve the metric significantly. In summary, mR@K is defined as

$$\textsc{Macro Recall@K} = \sum_{(s,p,o)\in\mathcal{T}_p} \frac{\textsc{Micro R@K}(p)}{|\mathcal{T}_p|} \tag{1.1}$$

1.3.4 Commonsense

While scene graphs or image captions describe *what is* in a specific image, *commonsense* or *semantic knowledge* is about *what can be* in an image (or the world in general) regardless of a given input. Note that commonsense could also refer to procedural and non-declarative forms of knowledge (e.g., how to drive a car [Sharifzadeh et al., 2016]) but the focus of this dissertation is on semantic knowledge that can be described by symbols [2] Symbolic knowledge can be represented in a structured form such as those in knowledge graphs (KGs) or unstructured such as the human language (e.g., encyclopedia, books, etc.). Since our language or knowledge graphs often have gaps in the information they contain, creating models of knowledge graphs or language is beneficial. Language and knowledge graph embedding models can generalize to new facts and help us predict the missing information, a task known as link prediction. We will briefly describe knowledge graph embeddings models and then discuss some recent language models.

Knowledge Graphs

Knowledge Graphs (KGs) are a data structure to store factual knowledge. A KG can be described as the following:

Definition 1.3.2 (Knowledge Graph)**.** *A Knowledge Graph (KG) is a 3-tuple* $\mathcal{K} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ *with a set of entities* $\mathcal{E}$*, a set of relations* $\mathcal{R}$*, and a set of triples* $\mathcal{T} \subseteq (\mathcal{E} \times \mathcal{R} \times \mathcal{E})$*.*

[2] Here, we use the term symbol, as one would do in the GOFAI (good old-fashioned artificial intelligence) methods instead of considering every possible continuous vector as a symbol.

Each triple $(h, r, t) \in \mathcal{T}$ with *head* h, *relation* r, and *tail* entity t, represent a fact that is present in our source of knowledge.

Knowledge Graphs vs. Scene Graphs

Note that knowledge graphs are defined the same as the scene graphs. The only difference is that SGs describe images, whereas KG can contain knowledge obtained from any modality or inferred through reasoning. Therefore, one can consider a set of several SGs as a single KG containing visual knowledge. The relation between KGs and SGs can help us define an interaction between them: (1) KGs can be built from SGs, and (2) SGs that are predicted from an image, can be directly compared with KGs; a predicted SG can be noisy or incomplete, because objects occlude each other or there are variations in lighting conditions, viewpoints, object poses, etc. As a result, vision models cannot correctly capture all the objects in the scene. However, KGs contain a more robust, general world structure and can fill in the missing gaps or correct errors. For example, if the back wheel of a car is not present in its image, the SG will not have it, but the KG can predict it.

Knowledge Graph Embedding Models

Knowledge Graphs are often incomplete or noisy since it is impossible to access all possible relations in the world. For example, a KG might contain a fact about giraffes having four legs, but not about *giraffe calves* having four legs. Knowledge Graph Embedding (KGE) models [Nickel et al., 2016] provide an efficient approach to infer these missing links (a task known as *link prediction*). KGE models assign a vector representation to each symbolic entity in the knowledge graph such that the interaction between the vector embeddings for head, tail, and predicate in a relation, can predict whether this fact is true or not. This can be considered a form of self-supervised learning where the input is a noisy graph (with dropped edges), and the goal is to reconstruct the graph. Once we train a knowledge graph embedding model, we can evaluate the generalization power of this model compared to the original symbolic KG. In our previous example, since a giraffe calf and its calf appear in similar relations in a KG, they will be given embeddings similar to each other but dissimilar to other entities. We can now predict the missing relations that a giraffe or its baby can have through the vector interactions. Therefore, instead of querying a knowledge graph for a symbolic fact, one can query a knowledge graph embedding by evaluating the composition of vector representations assigned to each symbol.

To find the embedding for each symbol in the knowledge graphs, the first knowledge graph embedding model, RESCAL [Nickel et al., 2011] proposed a method based on the expectation maximization algorithm [Dempster et al., 1977]. Nowadays, most link prediction methods, including variations of RESCAL, use artificial neural networks and gradient-based learning. There are several different KGE models; the main difference is in how they define the interaction function between vectors. Given the embeddings for head, predicate, and tail, the interaction function predicts whether this relation exists or not, such that it output is 1 for true relations and 0 otherwise. For example, RESCAL [Nickel et al., 2011] defines each relation as an affine transformation in the embedding space of entities. Therefore, the interaction function f is

$$f(\mathbf{x}_h, \mathbf{X}_r, \mathbf{x}_t) = \mathbf{x}_h^T \mathbf{X}_r \mathbf{x}_t$$

with $\mathbf{X}_r \in \mathbb{R}^{d \times d}$ as the d-dimensional matrix, representing relation r, $\mathbf{x}_h \in \mathbb{R}^{d_e}$ as the embedding for head entity and $\mathbf{x}_t \in \mathbb{R}^{d_e}$ as the embedding for the tail entity.

TransE [Bordes et al., 2013] defines the relation as a simple translation in space with the interaction function

$$f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t) = -\|\mathbf{x}_h + \mathbf{x}_r - \mathbf{x}_t\|$$

where $\mathbf{x}_r \in \mathbb{R}^{d_r}$ represents relation r. Compared to RESCAL, TransE has fewer parameters but cannot model symmetric relations.

DistMult [Yang et al., 2014] considers each relation as a vector, minimizes the trilinear dot product of subject, predicate, and object vector, and can be thought of as a form of RESCAL, where the transformation matrix is diagonal such that

$$f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t) = \langle \mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t \rangle$$

where $\langle \cdot, \cdot, \cdot \rangle$ denotes the tri-linear dot product.

ComplEx [Trouillon et al., 2016] is similar to DistMult, but in the complex space, therefore the interaction function is

$$f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t) = \Re\left(\langle \mathbf{x}_h, \mathbf{x}_r, \overline{\mathbf{x}_t} \rangle\right)$$

where $\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t \in \mathbb{C}^d$, $\Re$ denotes the operation which retrieves the real part of a complex number, and $^-$ the complex conjugate.

RotatE [Sun et al., 2019] models the relations as rotations in the complex space as

$$f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t) = -\|\mathbf{x}_h \odot \mathbf{x}_r - \mathbf{x}_t\|_2$$

with $\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t \in \mathbb{C}^d$, the relation representation being element-wise normalized to unit length $|(\mathbf{x}_r)_i| = 1$.

Triple-input [Dong et al., 2014] multilayer perceptron (MLP) architectures extends these methods to non-linear transformations with the interaction function

$$f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t) = \text{MLP}([\mathbf{x}_h; \mathbf{x}_r; \mathbf{x}_t]),$$

where MLP is a trained 2-layer Multi-Layer Perceptron (MLP).

Dual-input MLP used in Sharifzadeh et al. [2020] slightly modifies the previous interaction model and proposes the interaction function

$$f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t) = \mathbf{x}_t^T \text{MLP}([\mathbf{x}_h; \mathbf{x}_r]).$$

For an extensive review and study on different KG models, refer to [Nickel et al., 2016, Ali et al., 2020, 2021].

Language Models

Human language is a vast source of knowledge; the sentences communicated orally between humans or those written down, describe facts that are observed directly in the world or inferred through reasoning. Therefore, similar to KGs, language can give us a more general knowledge of the world that can help to correct the errors of a vision model. However, compared to the knowledge graphs, language is unstructured; there are symbols in the language that do not carry any information (e.g. "a"), or some entities that are referred to using different symbols (e.g. "Obama" became the president. "He" moved to the White House.). This makes it hard to query texts directly (e.g., Where is Obama?). On the other hand, unstructured texts are much more widely available worldwide than structured KGs, and exploiting this knowledge can be quite valuable.

Similar to how KGEs model knowledge graphs, one can model the language using language models (LMs). Given the words $\{w_1, w_2, ..., , w_n\}$ in a sentence of length n, a language model assigns a probability distribution $P(w_1, w_2, ..., , w_n)$ to the sentence. While modeling languages has a long history in computational linguistics with various methods, we focus on modern language models based on artificial neural networks, specially

transformers [Vaswani et al., 2017], such as BERT [Devlin et al., 2018], T5 [Raffel et al., 2019], GPT-3 [Brown et al., 2020], and Chinchilla [Hoffmann et al., 2022] with strong capabilities in modeling languages. These methods often rely on self-supervised training objectives such that their goal is to predict a the next tokens given previous ones, or a randomly masked version of the input. Similar to KGs, the knowledge stored in texts can be incomplete, and language models can enable us to infer unseen facts.

Language models have shown strong capabilities in many tasks such as translation, reasoning, etc. In this dissertation, we show that a pre-trained large language model (T5) can also be used to convert unstructured texts into knowledge graphs which can then be used in scene graph classification.

Since most of the modern language models are based on transformers, and we also use transformers on the graph structures in this dissertation, we briefly describe the graph transformers.

Transformers Bahdanau et al. [2014] proposed an attention mechanism originally for machine translation and to allow the model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word. Later, Vaswani et al. [2017] proposed multi-headed self-attentions in the transformer architecture to encode a sequence of tokens where each token considers the representation of itself and all the other neighbors. While the original transformers focus on sequences of sentences, one can apply transformers in a more general form, similar to graph convolutions [Kipf and Welling, 2016], that propagates messages between nodes and edges in a graph. In this work, we often use transformers in the more general graphical form that can be applied to objects and relations from images. To give a broad overview, consider an initial node embeddings $\mathbf{z}_i^{(0)}$ of node i in the first layer of a transformer; we compute $\mathbf{z}_i^{(l)}$ of the l-th transformer layer by computing and propagating messages.

$$\mathbf{m}_i^{\mathcal{N}(i)} = \frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l,k)} \mathbf{W}^{(l,k)} \mathbf{z}_j^{(l)} \tag{1.2}$$

$$\mathbf{z'}_i^{(l)} = LN(\mathbf{z}_i^{(l)} + \mathbf{m}_i^{\mathcal{N}_{in}(i)} + \mathbf{m}_i^{\mathcal{N}_{out}(i)}) \tag{1.3}$$

$$\mathbf{z}_i^{(l+1)} = LN(\mathbf{z'}_i^{(l)} + f(\mathbf{z'}_i^{(l)})), \tag{1.4}$$

where LN is the layer norm [Ba et al., 2016], K is the number of attentional heads and $\mathbf{W}^{(l,k)}$ is the weight matrix of the k-th head in layer l. $\mathcal{N}(i)$ represent the set of neighbors, which are either incoming $\mathcal{N}_{in}(i)$ or outgoing $\mathcal{N}_{out}(i)$. $f(.)$ is a two-layered feedforward neural network with Leaky ReLU non-linearities between each layer. $\alpha_{ij}^{(l,k)}$ denotes the attention coefficients in each head and is defined as

$$e_{ij}^{(l,k)} = \sigma(\mathbf{h}^{(l,k)} \cdot [\mathbf{z}_i^{(l)} || \mathbf{W}^{(l,k)} \mathbf{z}_j^{(l)}]) \tag{1.5}$$

$$\alpha_{ij}^{(l,k)} = \frac{\exp(e_{ij}^{(l,k)})}{\sum_{q \in \mathcal{N}(i)} \exp(e_{iq}^{(l,k)})} \tag{1.6}$$

with $\mathbf{h}^{(l,k)}$ as a learnable weight vector and $||$ denoting concatenation. σ is the Leaky ReLU with the slope of 0.2.

In the original transformers, a positional encoding is added to each embedding so that the model can consider the order of the words in a sentence. In our methods, we also use a form of spatial vector to encode the location of each node (object) in an image. In the end, the output of the final transformer layer is used for the task at hand, e.g., node or edge classification.

1.3.5 Visual Language Models

As discussed in the previous section, most language models take the input words $\{w_1, w_2, ..., , w_n\}$ and assign it a probability distribution $P(w_1, w_2, ..., , w_n)$. Visual language models (VLM) function similarly, except that the input can be a combination of word tokens ($\{w_1, w_2, ..., , w_n\}$) from the sentence *and* visual tokens ($\{v_1, v_2, ..., , v_n\}$) from image(s) or videos. This allows the visual language models to do different tasks, including visual question answering and captioning. In one of the most recent VLMs, Frozen [Tsimpoukelli et al., 2021], the penultimate layer of a convolutional neural network is directly fed into a pre-trained and frozen language model such that it generates captions for the given image inputs by modeling $P(v_1, v_2, ..., v_n, w_1, w_2, ..., w_n)$. This way, Frozen reduces the burden of modeling the language itself, and instead, the training can focus on learning to extract the most relevant visual features. Another benefit of having a pre-trained language model is that the generated captions become very rich in details, considering that the language model can already generate highly complex sentences.

More recently, in Flamingo [Alayrac et al., 2022] we showed that one could combine

pre-trained, frozen "vision-only" and "language-only" models and train a few adaptation layers only. This way, we need much less parallel text-image data to train the VLMs. Flamingo also showed that it can handle arbitrarily interleaved visual and textual data sequences by modeling $P(v_1, w_2, ..., v_n, w_1, v_5, ..., w_n)$. As a result, a user can communicate with the model in a multimodal way at test time. As shown in Sharifzadeh et al. [2021] and Sharifzadeh et al. [2022], combining knowledge graphs with scene graphs has similar benefits as modern VLMs for incorporating commonsense knowledge with vision models, albeit with structured outputs rather than texts.

1.3.6 The Symbol Grounding Problem

As discussed in the introduction, the symbol grounding problem is one of the core topics in this dissertation. According to Harnad [Harnad, 1990], the symbol grounding problem explores the question "Where do symbols get their meanings?". However, why is it crucial to study the symbol grounding problem? In order to understand the importance of grounding symbols from a computational perspective, let us consider the example of knowledge graphs as a representation of structured symbolic facts. As discussed earlier, knowledge graphs consist of triples indicating the relation between symbols, and a knowledge graph embedding (KGE) model assigns high-dimensional vector representations to each symbol. It might be tempting to consider this process of assigning an embedding to a symbol as a form of grounding by arguing that the relational structure of a graph gives rise to the meaning of symbols. However, while KGEs can help predict novel facts, they are still merely a generalization from symbols, and their embedding does not ground us in the real world. Harnad expresses this well in his example of the Chinese/Chinese Dictionary-Go-Round Problem [Harnad, 1990]: "Suppose you had to learn Chinese as a second language and the only source of information you had was a Chinese/Chinese dictionary. The trip through the dictionary would amount to a merry-go-round, passing endlessly from one meaningless symbol or symbol-string (the definientes) to another (the definienda), never coming to a halt on what anything meant".

The human brain seems to circumvent this issue by grounding symbols in perceptions [Piaget, 1923]. This gives rise not only to meaning but also to more substantial generalization; let us say that we have triples describing the relations between *Man*, *Woman*, and *Horse*. However, none of these triples can adequately describe the visual appearances of these entities. Therefore, these models fall short of predicting some relations. For example, assume that a man has seen and rode on horses in his life. Since

a horse and a donkey look similar, when he sees a donkey for the first time, it is quite straightforward for him to guess that he can also ride the donkey. However, the KGEs cannot predict this relation since they do not have access to any perceptual input and can only generalize within declarative relations. In order to include that in our model, we should also include higher-dimensional attributes, connecting the node of *horse* and *donkey* to their observed images (note that this image can be already in an embedding space rather than the original input space). This will ground the symbols not just in the relational structure of the symbolic graph but also in visual attributes.

Later in this dissertation, we show that grounding symbol representations into perceptions not only leads to better model generalization but also enables multimodal knowledge acquisition [Sharifzadeh et al., 2021, 2022].

2 Classification by attention: Scene graph classification with prior knowledge

This chapter comprises the publication

Sharifzadeh et al. [2021]

and the code is available at

`https://github.com/sharifza/schemata`

Declaration of Authorship The research idea was developed and conceptualized by Sahand Sharifzadeh. Sahand Sharifzadeh did the main part of the model implementation and design of the experiments. Sina Baharlou contributed to parts of the pre-processing and evaluation code, and taking care of the data splits. The final manuscript was mainly written by Sahand Sharifzadeh and revised by all authors.

Classification by Attention: Scene Graph Classification with Prior Knowledge

Sahand Sharifzadeh,[1] **Sina Moayed Baharlou,**[1*] **Volker Tresp**[1,2]
[1]Ludwig Maximilian University of Munich
[2]Siemens AG

sharifzadeh@dbs.ifi.lmu.de

Abstract

A major challenge in scene graph classification is that the appearance of objects and relations can be significantly different from one image to another. Previous works have addressed this by relational reasoning over all objects in an image or incorporating prior knowledge into classification. Unlike previous works, we do not consider separate models for perception and prior knowledge. Instead, we take a multi-task learning approach by introducing *schema* representations and implementing the classification as an attention layer between image-based representations and the schemata. This allows for the prior knowledge to emerge and propagate within the perception model. By enforcing the model also to represent the prior, we achieve a strong inductive bias. We show that our model can accurately generate commonsense knowledge and that the iterative injection of this knowledge to scene representations, as a top-down mechanism, leads to significantly higher classification performance. Additionally, our model can be fine-tuned on external knowledge given as triples. When combined with self-supervised learning and with 1% of annotated images only, this gives more than 3% improvement in object classification, 26% in scene graph classification, and 36% in predicate prediction accuracy.

Introduction

Classifying objects and their relations in images, also known as scene graph classification, is a fundamental task in scene understanding and can play an essential role in applications such as recommender systems, visual question answering and decision making. Scene graph (SG) classification methods typically have a perception model that takes an image as input and generates a graph that describes the given image as a collection of (`head, predicate, tail`). One of the main challenges that current models face is diverse appearances of objects and relations across different images. This can be due to variations in lighting conditions, viewpoints, object poses, occlusions, etc. For example, the `Bowl` in Figure 1 is highly occluded and has very few image-based features. Therefore, a typical perception model fails to classify it. One approach to tackle this problem is to collect supportive evidence from the neighbors before classifying an entity. This

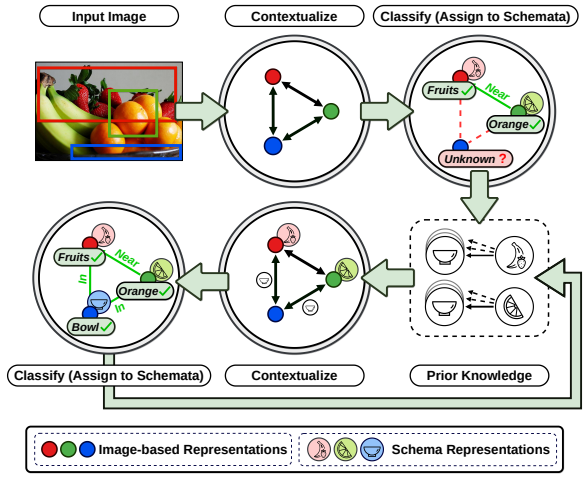

Figure 1: An example of scene graph classification where the Bowl lacks sufficient visual input. The top right is the initially predicted graph from the visual inputs only. The bottom left is the prediction of our model after considering both image-based representations and the prior knowledge about Fruits and Oranges (schemata). The long arrow near the bottom indicates recursion.

can be done, for example, by message passing between all the image-based object representations in an image, using graph convolutional neural networks (GCN) (Kipf and Welling 2016) or LSTMs (Hochreiter and Schmidhuber 1997). The main issue with this approach is the combinatorial explosion of all possible *image-based* neighbor representations[1].

A current theory in cognitive psychology states that humans solve this challenge by reasoning over the pre-existing representations of neighboring objects instead of relying on the perceptual inputs only (Piaget 1923); philosophers often argue that humans have a form of mental representation for objects and concepts (Kant 1787). These representations do not depend on a given image but are rather *symbol-based*. There are different opinions on how these representations come to be. Piaget called these representations *schema* (plu-

[*]S.M. Baharlou contributed to this project while he was a visiting researcher at the Ludwig Maximilian University of Munich.

[1]For a more detailed probabilistic analysis of this issue, refer to the section *GCN vs. Prior Model: A matter of inductive biases.*

ral *schemata*) and suggested that we acquire them in our earlier perceptions. When an object is being perceived, the mind assigns it to a schema in a process called *assimilation*. By relational reasoning over schemata, assimilation helps to predict the facts surrounding the observation (Arbib 1992)[2].

Nevertheless, learning and utilizing prior knowledge is still a significant challenge in AI research. Recently, Zellers et al. (2018) and Chen et al. (2019c,b) proposed to correct the SG prediction errors by comparing them to the co-occurrence statistics of triples in the training dataset or an external source. The statistics can, for example, suggest that it is common to see `(Fruits, in, Bowls)`. Furthermore, instead of relying on simple co-occurrence statistics, one can create a prior model with knowledge graph embeddings (KGE) (Nickel et al. 2016) that can generalize beyond the given triples (Baier, Ma, and Tresp 2017, 2018; Hou et al. 2019). KGEs typically consist of an *embedding matrix*, that assigns a vector representation to each entity, and an *interaction function* that predicts the probability of a triple given the embeddings of its head, predicate, and tail. This allows KGE models to generalize to unseen relations. For example, `Man` and `Woman` will be given similar embeddings since they appear in similar relations during the training. As a result, if the model observes `(Woman, rides, Horse)`, it can generalize, for example, to `(Man, rides, Camel)`.

However, in the described approaches, unlike Piaget's schemata, the perception and the prior are treated naively as independent components; they are trained separately and from different inputs (either from images or from triples), and their predictions are typically fused in the probability space, e.g., by multiplication. Other than requiring redundant parameters and computations, this makes the prior model agnostic to the image-based knowledge and the perception model agnostic to the prior knowledge. For example, the collection of triples might contain `(Woman, rides, Horse)` but have no triples regarding a `Donkey`. While the images can represent the visual similarities of a `Horse` to a `Donkey`, the triples lack this information. If we train a prior model purely based on the triples, the model fails to generalize. We can avoid this by training the prior model from a combination of triples and images. As for another example, in Figure 1, the prior model might suggest `(Fruits, in, Bowl)` but it might also suggest `(Fruits, areOn, Trees)`. To decide between the two, one should still consider the visual cues from the given image.

To address these shortcomings, we entangle the perception and prior in a single model with shared parameters trained by multi-task learning. Therefore, instead of training a separate embedding matrix for a prior model, we exploit the perception model's classification layer; when we train a classification layer on top of contextualized image-based representations, the classification weights capture both relational and image-based class embeddings (Refer to Figure 2). Unfortunately, the classification's common realization as a fully connected layer does not allow us to feed these *network weights* to an interaction function. To this end, we employ a more general formulation of classification as an attention

[2]We leave out the discussion on the schemata in Kant's view.

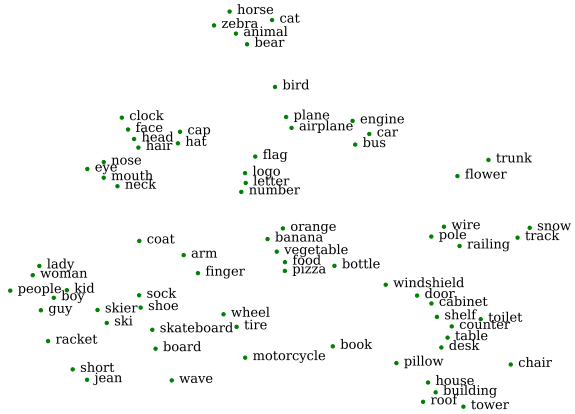

Figure 2: t-SNE visualization of the object classification weights that have been trained on top of contextualized image-based representations. Entities that appear similar to each other, or participate in similar relations, have a closer semantic affinity. This enables link prediction similar to Knowledge Graph Embeddings, and leads to generalization.

layer instead. In this layer, the extracted image-based and contextually enriched representations attend to trainable schema embeddings of all classes such that (a) the attention coefficients are the classification scores (we enforce this by applying a classification loss on the attention outputs), and (b) the attention values carry the prior knowledge that is injected into the image-based representations (a top-down mechanism).

Furthermore, instead of training a separate interaction function for the prior model, we exploit the message passing function that we already have available in the perception model; after fusing the schemata and the image-based object-representations, we contextualize and classify the representations again. Other than computational efficiency, this has the advantage that the image-based object representations and the schemata are combined in the embedding space rather than the probability space.

We train the schemata using the Visual Genome (Krishna et al. 2017) dataset. We show that our model can accurately generate the captured commonsense knowledge and that iterative injection of this knowledge, as a top-down signal, leads to significantly higher classification accuracy. Additionally, we draw from the recent advancements in self-supervised learning and show that the schemata can be trained with only a fraction of labeled images. This allows us to fine-tune the perception model without any additional images; instead, we can use a knowledge base of hand-crafted or external triples and train with their *mental images* (schemata). As a result, compared to the self-supervised baseline, and with 1% of the training data, our model achieves more than 3% improvement in object classification, 26% in scene graph classification, and 36% in predicate prediction; an accuracy that is almost equal to when using 100% of the labeled images.

Related Works

While the concept of schemata can be applied to any form of perceptual processing, and there are recent deep learning

models of action schemata (Kansky et al. 2017), we focus on the figurative schemata in the visual scene understanding domain. There has also been a body of related research on relational reasoning outside the scene graph domain (Wu, Lenz, and Saxena 2014; Deng et al. 2014; Hu et al. 2016, 2017; Santoro et al. 2017; Sabour, Frosst, and Hinton 2017). Nevertheless, research in this field was largely accelerated after the release of Visual Relation Detection (VRD) (Lu et al. 2016) and the Visual Genome (Krishna et al. 2017) datasets. Baier, Ma, and Tresp (2017, 2018) proposed the first KG-based model of prior knowledge that improves SG classification. VTransE (Zhang et al. 2017) proposed to capture relations by applying the KGE model of TransE (Bordes et al. 2013) on the visual embeddings. Yu et al. (2017) employed a teacher-student model to distill external language knowledge. Iterative Message Passing (Xu et al. 2017), Neural Motifs (Zellers et al. 2018) (NM), and Graph R-CNN (Yang et al. 2018) used RNNs and graph convolutions to propagate image context. Tang et al. (2019) exploited dynamic tree structures and Chen et al. (2019a) proposed a method based on multi-agent policy gradients. Sharifzadeh et al. (2019) employed the predicted pseudo depth maps in addition to the 2D information. In general, scene graph classification methods are closely related to KGE models (Nickel, Tresp, and Kriegel 2011; Nickel et al. 2016). For an extensive discussion on the connection between perception, KGEs, and cognition, refer to (Tresp, Sharifzadeh, and Konopatzki 2019; Tresp et al. 2020). The link prediction in KGEs arises from the compositionality of the trained embeddings. Some other forms of compositionality in neural networks are discussed in (Montufar et al. 2014). In this work, we introduce assimilation, which strengthens the representations within the neural network's causal structure, addressing an issue raised by Fodor, Pylyshyn et al. (1988). Some of the issues that we address in this work have also been recently discussed by Bengio (2017); Goyal et al. (2019); Mittal et al. (2020).

Methods

In summary, after an initial classification step, we combine the image-based representations with the schemata of predicted classes. We then collect supportive evidence from the neighbors before re-classifying each entity (Ref. Figure 1). In what follows, bold lower case letters denote vectors, bold upper case letters denote matrices, and the letters denote scalar quantities or random variables. Subscripts and superscripts denote variables and calligraphic upper case letters for sets.

Definitions

Let us consider a given image $\mathbf{I}$ and a set of n bounding boxes $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^n$, $\mathbf{b}_i = [b_i^x, b_i^y, b_i^w, b_i^h]$, such that $[b_i^x, b_i^y]$ are the coordinates of $\mathbf{b}_i$ and $[b_i^w, b_i^h]$ are its width and height. We build a **Scene Representation Graph**, *SRG* $= \{\mathcal{V}, \mathcal{E}\}$ as a structured presentation of the objects and predicates in $\mathbf{I}$. $\mathcal{X}^o = \{\mathbf{x}_i^o\}_{i=1}^n$, $\mathbf{x}_i^o \in \mathbb{R}^d$ denote the features of object nodes and $\mathcal{X}^p = \{\mathbf{x}_i^p\}_{i=1}^m$, $\mathbf{x}_i^p \in \mathbb{R}^d$ denote the features of predicate nodes[3]. Each $\mathbf{x}_i^o$ is initialized by a pooled image-based object representation, extracted by applying VGG16 (Simonyan and Zisserman 2014) or ResNet-50 (He et al. 2016) on the image contents of $\mathbf{b}_i$. Each $\mathbf{x}_i^p$ is initialized by applying a two layered fully connected network on the relational position vector $\mathbf{t}$ between a head i and a tail j where $\mathbf{t} = [t_x, t_y, t_w, t_h]$, $t_x = (b_i^x - b_j^x)/b_{i\ j}^w$, $t_y = (b_i^y - b_j^y)/b_j^h, t_w = \log(b_i^w/b_j^w), t_h = \log(b_i^h/b_j^h)$. The implementation details of the networks are provided in the Supplementary.

Scene graph classification is the mapping of each node in scene representation graph to a label where each object node is from the label set $\mathcal{C}^o$ and each predicate node from $\mathcal{C}^p$. The resulting labeled graph is a set of triples referred to as the **Scene Graph**. We also define a **Probabilistic Knowledge Graph** (*PKG*) as a graph where the weight of a triple is the expected value of observing that relation given the head and tail classes and regardless of any given images[4]. Later we will show that our model can accurately generate the PKG, i.e., the commonsense that is captured from perceptions during training.

In what comes next, $\mathbf{x}_i^o$ and $\mathbf{x}_i^p$ are treated identically except for classification with respect to $\mathcal{C}^o$ or $\mathcal{C}^p$. Therefore, for a better readability, we only write $\mathbf{x}_i$.

Contextualized Scene Representation Graph

We obtain contextualized object representations $\mathbf{z}_i$ by applying a graph convolutional neural network, on *SRG*. We also refer to this module as our *interaction function*. We use a Graph Transformer as a variant of the Graph Network Block (Battaglia et al. 2018; Koncel-Kedziorski et al. 2019) with multi-headed attentions as

$$\mathbf{m}_i^{\mathcal{N}(i)} = \frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(l,k)}\mathbf{W}^{(l,k)}\mathbf{z}_j^{(l,t)} \tag{1}$$

$$\mathbf{z'}_i^{(l)} = \mathit{LN}(\mathbf{z}_i^{(l,t)} + \mathbf{m}_i^{\mathcal{N}_{in}(i)} + \mathbf{m}_i^{\mathcal{N}_{out}(i)}) \tag{2}$$

$$\mathbf{z}_i^{(l+1,t)} = \mathit{LN}(\mathbf{z'}_i^{(l)} + f(\mathbf{z'}_i^{(l)})), \tag{3}$$

where $\mathbf{z}_i^{(l,t)}$ is the embedding of node i in the l-th graph convolution layer and t-th assimilation. In the first layer $\mathbf{z}_i^{(0,t)} = \mathbf{x}_i$. *LN* is the layer norm (Ba, Kiros, and Hinton 2016), K is the number of attentional heads and $\mathbf{W}^{(l,k)}$ is the weight matrix of the k-th head in layer l. $\mathcal{N}(i)$ represent the set of neighbors, which are either incoming $\mathcal{N}_{in}(i)$ or outgoing $\mathcal{N}_{out}(i)$. $f(.)$ is a two layered feed-forward neural network with Leaky ReLU non-linearities between each layer. $\alpha_{ij}^{(l,k)}$ denotes the attention coefficients in each head and is defined as

$$e_{ij}^{(l,k)} = \sigma(\mathbf{h}^{(l,k)} \cdot [\mathbf{z}_i^{(l)} || \mathbf{W}^{(l,k)}\mathbf{z}_j^{(l)}]) \tag{4}$$

$$\alpha_{ij}^{(l,k)} = \frac{\exp(e_{ij}^{(l,k)})}{\sum_{q\in\mathcal{N}(i)}\exp(e_{iq}^{(l,k)})} \tag{5}$$

[3]Similar to (Yang et al. 2018; Koncel-Kedziorski et al. 2019), we consider each object node as direct neighbors with its predicate nodes and each predicate node as direct neighbors with its head and tail object nodes.

[4]Note that while typical knowledge graphs such as Freebase are based on object instances, given the nature of our image dataset, we focus on classes.

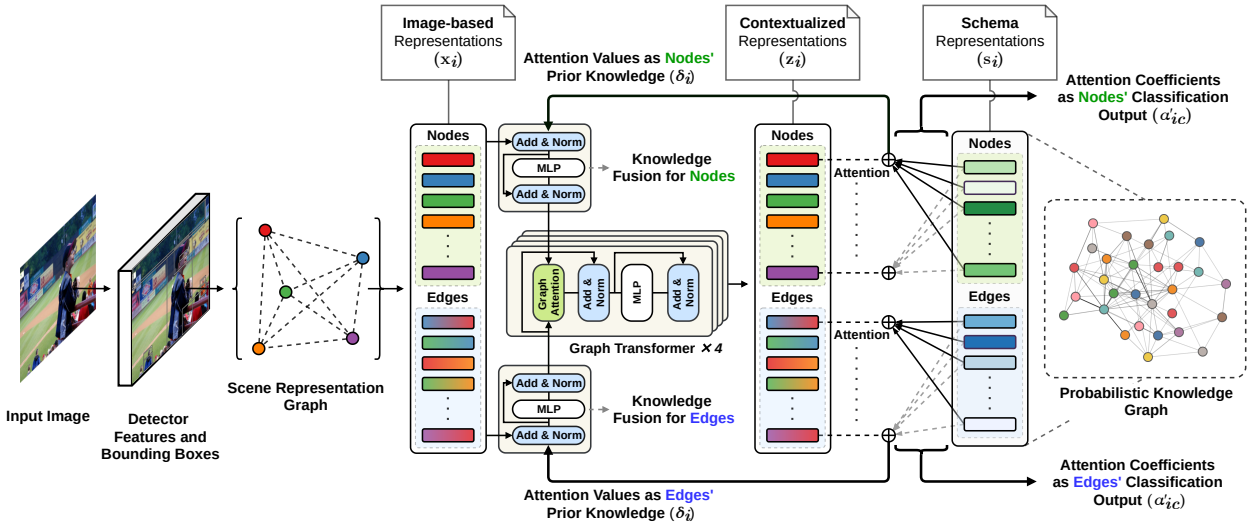

Figure 3: We formulate the classification as attention layer between object and schema representations. Contextualizing image-based object representations before classification encourages the schemata to learn *image-based relational* prior knowledge. As a result, the attention values that are injected from the schemata to scene representations and then propagated. In this way, they enrich the image-based representations with prior knowledge. Additionally, the interactions between schemata can reconstruct the probabilistic knowledge graph (right).

with $\mathbf{h}^{(l,k)}$ as a learnable weight vector and $||$ denoting concatenation. σ is the Leaky ReLU with the slope of 0.2.

Schemata

We define the schema of a class c as an embedding vector $\mathbf{s}_c$. We realize object and predicate classification by an attention layer between the contextualized representations and the schemata such that the classification outputs α'_{ic} are computed as the attention coefficients between $\mathbf{z}_i$ and $\mathbf{s}_c$ as

$$\alpha'_{ic} = \text{softmax}(a(\mathbf{z}_i^{(L,t)}, \mathbf{s}_c)) \tag{6}$$

where, $a(.)$ is the attention function that we implement as the dot-product between the input vectors, and $\mathbf{z}_i^{(L,t)}$ is the output from the last (L-th) layer of the Graph Transformer. The attention values $\boldsymbol{\delta}_i$ capture the schemata messages as

$$\boldsymbol{\delta}_i = \sum_{c \in \mathcal{C}} \alpha'_{ic} \mathbf{s}_c \tag{7}$$

and we inject them back to update the scene representations as

$$\mathbf{u}_i = LN(\mathbf{x}_i + \boldsymbol{\delta}_i) \tag{8}$$

$$\mathbf{z}_i^{(0,t+1)} = LN(\mathbf{u}_i + g(\mathbf{u}_i)) \tag{9}$$

where $g(.)$ is a two-layered feed-forward network with Leaky ReLU non-linearities. Note that we compute $\mathbf{u}_i$ by fusing the attention values with the *original image features* $\mathbf{x}_i$. Therefore, the outputs from previous Graph Transformer layers will not be accumulated, and the original image-based features will not vanish.

We define *assimilation* as the set of computations from $\mathbf{z}_i^{(L,t)}$ to $\mathbf{z}_i^{(L,t+1)}$. This includes the initial classification step (Eq. 6), fusion of schemata with image-based vectors (Eq. 9) and the application of the interaction function on the updated embeddings (Eq. 3). We expect to get refined object representations after the assimilation. Therefore, we assimilate several times such that after each update of the classification results, the priors are also updated accordingly. During training, and for each step of assimilation, we employ a supervised attention loss, i.e. categorical cross entropy, between the one-hot encoded ground truth labels and α'_{ic}. This indicates a multi-task learning strategy where one task (for the first assimilation) is to optimize for $P(y_q|x_1, ..., x_\theta)$, with x_q as a random variable representing the image-based features of q, y_q as the label, and $\theta = m + n$. The other set of tasks is to optimize for $P(y_q^{t+1}|x_1^t, ..., x_\theta^t, y_1^t, ..., y_\theta^t)$. We refer to the first task as **IC**, for **I**mage-based **C**lassification and to the second set of tasks as **ICP** for **I**mage-based **C**lassification with **P**rior knowledge. We train the second task using teacher forcing and by setting the labels to their ground truth values. Therefore, in order to prevent collapse, we set the edge schemata to zero. This resembles link prediction, such that we denoise an incomplete input graph. Note that even when no images are available, we can still train for the *ICP* from a collection of external or hand-crafted triples by directly assigning $\mathbf{z}_i^{(0,t+1)} = \boldsymbol{\delta}_i$ such that $\alpha'_{ic} = onehot(c_i)$.

GCN vs. Prior Model: A matter of inductive biases

Typical GCNs, such as the Graph Transformer, take the features derived from each bounding box as input, apply non-linear transformations and propagate them to the neighbors in the following layers. Each GCN layer consists of fully connected neural networks. Therefore, *theoretically* they can also model and propagate prior knowledge that is *not* visible

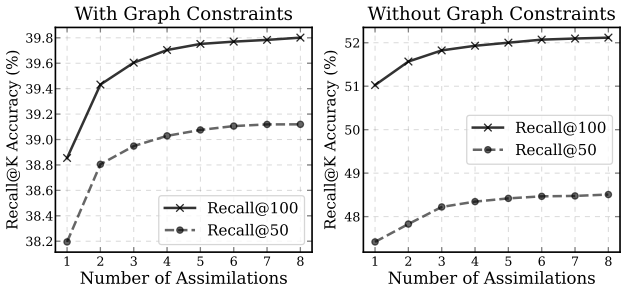

Figure 4: The results of our ablation studies. We study the effect of each assimilation in scene graph classification. Note that the model has been trained for only 4 assimilations yet it can generalize.

in bounding boxes. However, experimental results of previous works (and also this work) confirm that explicit modeling and propagation of prior knowledge (*ICP*) can still improve the classification accuracy. Why is that the case?

Let us consider the following. According to the the universal approximation theorem (Csáji et al. 2001), when we solve for *IC* as $P(y_k|x_1, ..., x_o)$, our model might learn to capture a desired form of $P(y_k|x_1, ..., x_o, y_1, ..., y_o)$. However, in practice, the learning algorithm does not always find the best function. Therefore, we require appropriate inductive biases to guide us through the learning process. As Caruana (1997) puts: *"Multitask Learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better"*. For example, in the encoder-decoder models for machine translation, e.g. Transformers (Vaswani et al. 2017), the prediction is often *explicitly* conditioned not just on the encoded inputs but also on the decoded outputs from the previous tokens. Therefore, the decoding in each step can be interpreted as computing $P(y_k|x_1, ..., x_o, y_1, ..., y_{k-1})$. Note that the previous predictions such as y_1, cannot benefit from the future predictions $\{y_2, ..., y_o\}$. However, in our model, we provide an explicit bias towards utilizing predictions in *all indices*. In fact, our model can be interpreted as an encoder-decoder network, where the decoder consists of multiple decoders. Therefore, the decoding depends not just on the encoded image features but also on the previously decoded outputs. In other words, by injecting schema embeddings, as embeddings that are trained over *all images*, we impose the bias to propagate *what is not visible in the bounding box*. As will be shown later, we can train for *ICP* and *IC* even with smaller splits of annotated images, which can lead to competitive results with fewer labels. Additionally, assimilation enables us to quantify the propagated prior knowledge. This interpretability is another advantage that GCNs alone do not have.

Settings We train our models on the common split of Visual Genome (Krishna et al. 2017) dataset containing images labeled with their scene graphs (Xu et al. 2017). This split takes the most frequent 150 object and 50 predicate classes in total, with an average of 11.5 objects and 6.2 predicates in each image. We report the experimental results on the test set, under two standard classification settings of predicate classification **(PredCls)**: predicting predicate labels given a ground truth set of object boxes and object labels, and scene graph classification **(SGCls)**: predicting object and predicate labels, given the set of object boxes. Another popular setting is the scene graph detection (SGDet), where the network should also detect the bounding boxes. Since the focus of our study is not on improving the object detector backbone and our improvements in SGDet were similar to the improvements in SGCls, we do not report them here. For those results, please refer to our official code repository. We report all the results under *constrained* and *unconstrained* setups (Yu et al. 2017). In the unconstrained setup, we allow for multiple predicate labels, whereas in the constrained setup, we only take the top-1 predicted predicate label.

	Method	SGCls		PredCls		Mean
		@50	@100	@50	@100	
Unconstrained	IMP+ (Xu et al. 2017)	12.1	16.9	20.3	28.9	19.5
	FREQ (Zellers et al. 2018)	13.5	19.6	24.8	37.3	23.8
	SMN (Zellers et al. 2018)	15.4	20.6	27.5	37.9	25.3
	KERN(Chen et al. 2019c)	19.8	26.2	36.3	49.0	32.8
	Schemata	**21.4**	**28.8**	**40.1**	**54.9**	**36.3**
Constrained	IMP (Xu et al. 2017)	3.1	3.8	6.1	8.0	5.2
	IMP+ (Xu et al. 2017)	5.8	6.0	9.8	10.5	8.0
	FREQ (Zellers et al. 2018)	6.8	7.8	13.3	15.8	10.9
	SMN (Zellers et al. 2018)	7.1	7.6	13.3	14.4	10.6
	KERN(Chen et al. 2019c)	9.4	10.0	17.7	19.2	14.0
	VCTree(Tang et al. 2019)	10.1	10.8	17.9	19.4	14.5
	Schemata	**10.1**	**10.9**	**19.1**	**20.7**	**15.2**
	Schemata - PKG	–.-	–.-	8.2	9.4	–.-

Table 1: Comparison of the mR@50 and mR@100, with and without graph constraints for SGCls and PredCls.

Metrics We use Recall@K (**R@K**) as the standard metric. R@K computes the mean prediction accuracy in each image given the top K predictions. In VG, the distribution of labeled relations is highly imbalanced. Therefore, we additionally report Macro Recall (Sharifzadeh et al. 2019; Chen et al. 2019c) (**mR@K**) to reflect the improvements in the long tail of the distribution. In this setting, the overall recall is computed by taking the mean over recall per predicate.

Experiments The goal of our experiments is **(A)** to study whether injecting prior knowledge into scene representations can improve the classification and **(B)** to study the commonsense knowledge that is captured in our model. In what follows, ***backbone*** refers to VGG16/ResNet-50 that generates the *SRG*, and ***main model*** refers to part of the network that applies contextualization and assimilation. The backbone can be trained from a set of labeled images (in a supervised manner), unlabeled images (in a self-supervised manner), or a combination of the two. The main model can be trained from

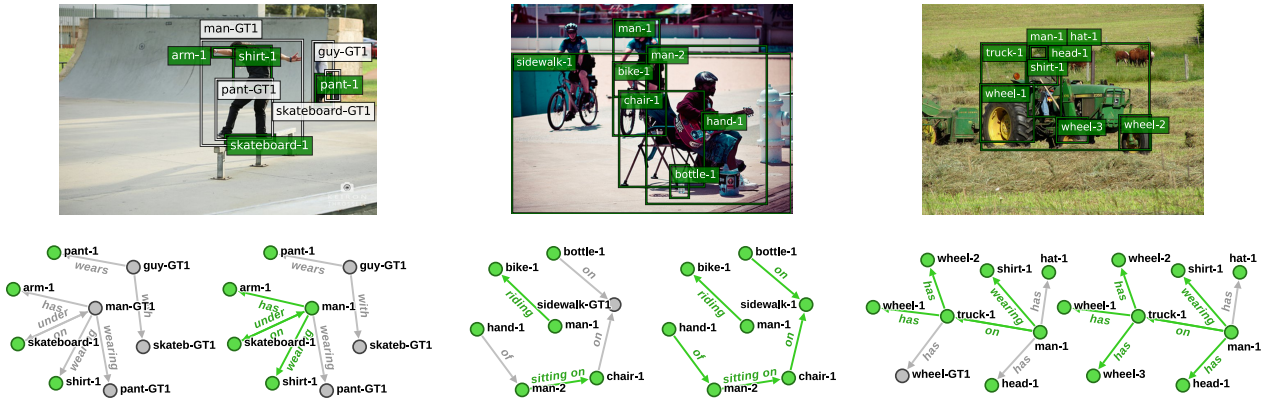

Figure 5: Qualitative examples of improved scene graph classification results (Recall@50) through assimilations of our model. From left to right is after each assimilation. Green and gray colors indicate true positives and false negatives concluded by the model. For example consider the middle image, where the sidewalk was initially misclassified as a street. After seeing a biker in the image and a man sitting on a chair, a reasonable inference is that this should be a sidewalk.

Training	SGCls R@100			PredCls R@100			Object Classification		
	1%	10%	100%	1%	10%	100%	1%	10%	100%
Sup - IC	1.84 ±0.26	13.90 ±0.97	33.6	40.61 ±0.84	52.51 ±1.19	62.0	14.38 ±0.57	38.45 ±1.21	64.2
Self-Sup - IC	12.12 ±0.47	26.14 ±0.77	36.8	48.10 ±0.54	58.14 ±0.35	63.4	40.75 ±0.48	56.97 ±0.76	68.0
Self-Sup - IC & ICP	**15.36** ±0.38	**27.37** ±0.47	**37.1**	**65.68** ±0.12	**65.42** ±0.19	**65.7**	**42.09** ±0.65	**58.60** ±0.56	**68.4**

Table 2: Comparison of R@100 for SGCls, PredCls and Object Classification tasks on smaller splits of the VG dataset.

a set of labeled images (the *IC* task), a prior knowledge base *(ICP)* or a combination of the two. For (**A**), we conduct the following studies:

1. We train both the backbone and the main model from all the labeled images and for both tasks. We use the VGG-16 backbone as trained by Zellers et al. (2018). This allows us to compare the results with the related works directly. We evaluate the classification accuracy for 8 assimilations (until the changes are not significant anymore). Table 1 compares the performance of our model to the state-of-the-art under mR@K (for the R@K results refer to the supplementary). As shown, our model exceeds the others on average and under most settings. supplementary. Figure 4 shows our ablation study, indicating that the accuracy is improved after each assimilation.
2. To qualitatively examine these results, we present some of the images and their scene graphs after two assimilations, in Figure 5. For example in the right image, while the wheel is almost fully occluded, we can still classify it once we classify other objects and employ commonsense (e.g., trucks have wheels). Another interesting example is the middle image, where the sidewalk is initially misclassified as a street. After seeing a biker in the image and a man sitting on a chair, a reasonable inference is that this should be a sidewalk! Similarly, in the left image, the man is facing away from the camera, and his pose makes it hard to classify him unless we utilize our prior knowledge about the arm, pants, shirt, and skateboard.
3. Figure 7 shows the improvements per each predicate class. The results indicate that most improvements occur in under-represented classes. This means that we have achieved a generalization performance that is beyond the simple reflection of the dataset's statistical bias.
4. To understand the importance of prior knowledge compared to having a large set of labeled images, we conduct the following study: we uniformly sample two splits with 1% and 10% of VG. The images in each split are considered as *labeled*. We ignore the labels of the remaining images and consider them as unlabeled[5]. Instead, we treat the set of ignored labels as a form of external/hand-crafted knowledge in the form of triples. For each split, we train the full model (I) with a backbone that has been trained in a supervised fashion with the respective split and no pre-training, and the main model that has been trained for *IC* (without commonsense) with the respective split, (II) with a backbone that has been pre-trained on ImageNet (Deng et al. 2009) and fine-tuned on the Visual Genome (in a self-supervised fashion with BYOL (Grill et al. 2020)) and fine-tuned on the respective split of the visual genome (in a supervised fashion) and the main model that has been trained for *IC* with the respective split, and (III) Similar to 2, except that we include the *ICP* and train the main model by assimilating the entire prior knowledge base including

[5]Note that these splits are different from the recently proposed few-shot learning set by Chen et al. (2019d). In (Chen et al. 2019d), the goal is to study the few-shot learning of *predicates* only. However, we explore a more competitive setting, where only a fraction of both *objects* and *predicates* are labeled.

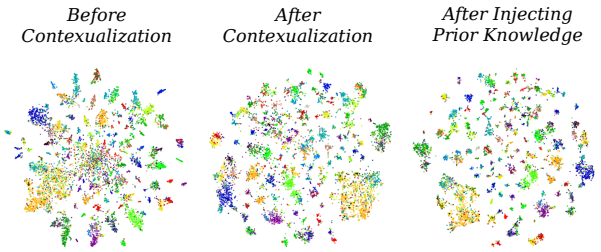

Figure 6: t-SNE visualization of object representations.

the external triples. We discard their image-based features ($\mathbf{x}_i$) for the triples outside a split. Also, to treat all triples equally when injecting the prior knowledge, we discard *all* image-based features and directly feed the $\boldsymbol{\delta}_i$ to the graph transformer. To prevent collapse, we randomly drop some of the $\boldsymbol{\delta}_i$s. Since BYOL is based on ResNet-50, for a fair comparison, we train all models in this experiment with ResNet-50 (including another model that we train with 100% of the data). In the Scene Graph Classification community, the results are often reported under an arbitrary random seed, and previous works have not reported the summary statistics over several runs before. To allow for a fair comparison of our model to those works (on the 100% set), we followed the same procedure in the study **A1**. However, to encourage a statistically more stable comparison of future models in this experiment, we report the summary statistics (arithmetic mean and standard deviation) over five random fractions (1% and 10%) of VG training set[6]. As shown in Table 2, utilizing prior knowledge allows to achieve almost the same predicate prediction accuracy with 1% of the data only. Also, we largely improve object classification and scene graph classification.

Evaluation

For **B** we consider the following studies:

1. We visualize the semantic affinity of *schema representations* by employing t-SNE (Maaten and Hinton 2008). As we can see in Figure 2, the schema representations of entities that are *visually* or *relationally* similar are the closest to each other.
2. We inspect the semantic affinity of *object representations* by employing t-SNE (I) before contextualization, (II) after contextualization and (III) after injecting prior knowledge. The results are represented in Figure 6. Each color represents a different object class. This investigation confirms that object representations will get into more separable clusters after injecting prior knowledge.
3. Finally, we evaluate our model's accuracy in link prediction. The goal is to quantitatively evaluate our model's understanding of *relational* commonsense, i.e., relational structure of the probabilistic knowledge graph. Similar to a KGE link prediction, we predict the predicate given *head* and *tail* of a relation. In other words, we feed our model with the schema of head and tail, together with a *zero-vector* for the image-based representations. As we can see in Table 1, in *Schemata - PKG*, even if we do not provide any image-based information, our model can still *guess* the expected predicates similar to a KGE model. While this guess is not as accurate as when we present it with an image, the accuracy is still remarkable.

[6]The splits are available at: https://github.com/sharifza/schemata

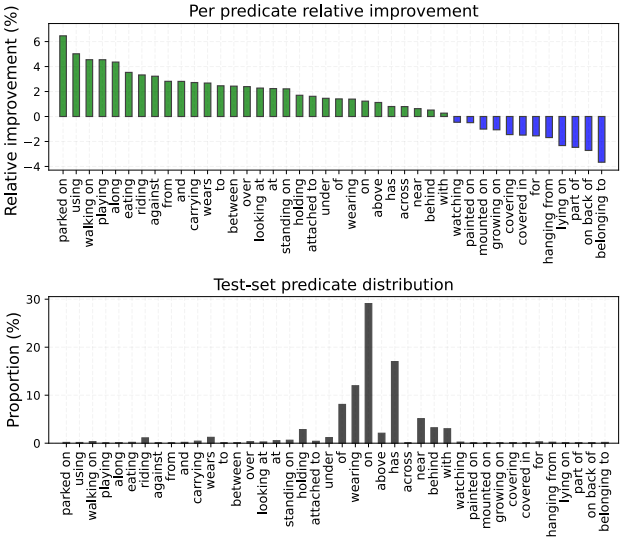

Figure 7: The top shows the per-predicate classification accuracy improvement after injecting prior knowledge, in SGCls R@100. The bottom shows the distribution of sample proportion for the predicates in the VG.

Conclusion

We discussed schemata as mental representations that enable compositionality and reasoning. To model schemata in a deep learning framework, we introduced them as representations that encode image-based and relational prior knowledge of objects and predicates in each class. By defining classification as an attention layer instead of a fully connected layer, we introduced an inductive bias that enabled the propagation of prior knowledge. Our experiments on the Visual Genome dataset confirmed the effectiveness of assimilation through qualitative and quantitative measures. Our model achieved higher accuracy under most settings and could also accurately predict the commonsense knowledge. Additionally, we showed that our model could be fine-tuned from external sources of knowledge in the form of triples. When combined with pre-trained schemata in a self-supervised setting, this leads to a predicate prediction accuracy that is almost equal to the full model. Also, it gives significant improvements in the scene graph and object classification tasks. We hope that this work will open new research directions in utilizing commonsense to learn from little annotations.

Acknowledgments

We would like to thank Max Berrendorf, Dario Konopatzki, Shaya Akbarinejad, Lisa Machata, Shabnam Sadegh, and the anonymous reviewers for the fruitful discussions and helpful feedback on the manuscript. This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A.

References

Arbib, M. A. 1992. Schema theory. *The encyclopedia of artificial intelligence* 2: 1427–1443.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Baier, S.; Ma, Y.; and Tresp, V. 2017. Improving visual relationship detection using semantic modeling of scene descriptions. In *International Semantic Web Conference*, 53–68. Springer.

Baier, S.; Ma, Y.; and Tresp, V. 2018. Improving information extraction from images with learned semantic models. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 5214–5218. AAAI Press.

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* .

Bengio, Y. 2017. The consciousness prior. *arXiv preprint arXiv:1709.08568* .

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Caruana, R. 1997. Multitask learning. *Machine learning* 28(1): 41–75.

Chen, L.; Zhang, H.; Xiao, J.; He, X.; Pu, S.; and Chang, S.-F. 2019a. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, 4613–4623.

Chen, T.; Xu, M.; Hui, X.; Wu, H.; and Lin, L. 2019b. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 522–531.

Chen, T.; Yu, W.; Chen, R.; and Lin, L. 2019c. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6163–6171.

Chen, V. S.; Varma, P.; Krishna, R.; Bernstein, M.; Re, C.; and Fei-Fei, L. 2019d. Scene graph prediction with limited labels. In *Proceedings of the IEEE International Conference on Computer Vision*, 2580–2590.

Csáji, B. C.; et al. 2001. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary* 24(48): 7.

Deng, J.; Ding, N.; Jia, Y.; Frome, A.; Murphy, K.; Bengio, S.; Li, Y.; Neven, H.; and Adam, H. 2014. Large-scale object classification using label relation graphs. In *European conference on computer vision*, 48–64. Springer.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Fodor, J. A.; Pylyshyn, Z. W.; et al. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition* 28(1-2): 3–71.

Goyal, A.; Lamb, A.; Hoffmann, J.; Sodhani, S.; Levine, S.; Bengio, Y.; and Schölkopf, B. 2019. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893* .

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Hou, J.; Wu, X.; Qi, Y.; Zhao, W.; Luo, J.; and Jia, Y. 2019. Relational Reasoning using Prior Knowledge for Visual Captioning. *arXiv preprint arXiv:1906.01290* .

Hu, H.; Deng, Z.; Zhou, G.-T.; Sha, F.; and Mori, G. 2017. Labelbank: Revisiting global perspectives for semantic segmentation. *arXiv preprint arXiv:1703.09891* .

Hu, H.; Zhou, G.-T.; Deng, Z.; Liao, Z.; and Mori, G. 2016. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2960–2968.

Kansky, K.; Silver, T.; Mély, D. A.; Eldawy, M.; Lázaro-Gredilla, M.; Lou, X.; Dorfman, N.; Sidor, S.; Phoenix, S.; and George, D. 2017. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. *arXiv preprint arXiv:1706.04317* .

Kant, I. 1787. *Kritik der reinen Vernunft:[Hauptband]*. Walter de Gruyter.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Koncel-Kedziorski, R.; Bekal, D.; Luan, Y.; Lapata, M.; and Hajishirzi, H. 2019. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1904.02342* .

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123(1): 32–73.

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2016. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 852–869. Springer.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Mittal, S.; Lamb, A.; Goyal, A.; Voleti, V.; Shanahan, M.; Lajoie, G.; Mozer, M.; and Bengio, Y. 2020. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In *International Conference on Machine Learning*, 6972–6986. PMLR.

Montufar, G. F.; Pascanu, R.; Cho, K.; and Bengio, Y. 2014. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, 2924–2932.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1): 11–33.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, 809–816.

Piaget, J. 1923. *Langage et pensée chez l'enfant*. Delachaux et Niestlé.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829* .

Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, 4967–4976.

Sharifzadeh, S.; Moayed Baharlou, S.; Berrendorf, M.; Koner, R.; and Tresp, V. 2019. Improving Visual Relation Detection using Depth Maps. *arXiv preprint arXiv:1905.00966* .

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Tang, K.; Zhang, H.; Wu, B.; Luo, W.; and Liu, W. 2019. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6619–6628.

Tresp, V.; Sharifzadeh, S.; and Konopatzki, D. 2019. A Model for Perception and Memory. *Conference on Cognitive Computational Neuroscience* .

Tresp, V.; Sharifzadeh, S.; Konopatzki, D.; and Ma, Y. 2020. The Tensor Brain: Semantic Decoding for Perception and Memory. *arXiv preprint arXiv:2001.11027* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wu, C.; Lenz, I.; and Saxena, A. 2014. Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception. In *Robotics: Science and systems*.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5419.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 670–685.

Yu, R.; Li, A.; Morariu, V. I.; and Davis, L. S. 2017. Visual relationship detection with internal and external linguistic knowledge distillation. In *IEEE International Conference on Computer Vision (ICCV)*.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5831–5840.

Zhang, H.; Kyaw, Z.; Chang, S.; and Chua, T. 2017. Visual Translation Embedding Network for Visual Relation Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 3107–3115. IEEE Computer Society. ISBN 978-1-5386-0457-1. doi:10.1109/CVPR.2017.331. URL https://doi.org/10.1109/CVPR.2017.331.

3 Improving Scene Graph Classification by Exploiting Knowledge from Texts

This chapter comprises the publication

Sharifzadeh et al. [2022]

and the code is available at

`https://github.com/mnschmit/unsupervised-graph-text-conversion`

`https://github.com/sharifza/schemata`

Declaration of Authorship The research idea was developed and conceptualized by Sahand Sharifzadeh. Sahand Sharifzadeh and Sina Baharlou did the main part of the model implementation and design of the experiments and extracted the data splits. Martin Schmitt implemented and evaluated the text-to-graph model. The final manuscript was mainly written by Sahand Sharifzadeh and revised by all authors.

The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

Improving Scene Graph Classification by Exploiting Knowledge from Texts

Sahand Sharifzadeh[1*], **Sina Moayed Baharlou**[1*†], **Martin Schmitt**[2], **Hinrich Schütze**[2], **Volker Tresp** [1,3]

[1] Department of Informatics, LMU Munich, Germany
[2] Center for Information and Language Processing (CIS), LMU Munich, Germany
[3] Siemens AG, Munich, Germany
sahand.sharifzadeh@gmail.com, sina.baharlou@gmail.com

Abstract

Training scene graph classification models requires a large amount of annotated image data. Meanwhile, scene graphs represent relational knowledge that can be modeled with symbolic data from texts or knowledge graphs. While image annotation demands extensive labor, collecting textual descriptions of natural scenes requires less effort. In this work, we investigate whether textual scene descriptions can substitute for annotated image data. To this end, we employ a scene graph classification framework that is trained not only from annotated images but also from symbolic data. In our architecture, the symbolic entities are first mapped to their correspondent image-grounded representations and then fed into the relational reasoning pipeline. Even though a structured form of knowledge, such as the form in knowledge graphs, is not always available, we can generate it from unstructured texts using a transformer-based language model. We show that by fine-tuning the classification pipeline with the extracted knowledge from texts, we can achieve ∼8x more accurate results in scene graph classification, ∼3x in object classification, and ∼1.5x in predicate classification, compared to the supervised baselines with only 1% of the annotated images.

Introduction

Relational reasoning is one of the essential components of intelligence; humans explore their environment by grasping the entire context of a scene rather than studying each item in isolation from the others. Furthermore, we expand our understanding of the world by educating ourselves about novel facts through reading or listening. For example, we might have never seen a "cow wearing a dress" but might have read about Hindu traditions of decorating cows. While we already have a robust visual system that can extract basic visual features such as edges and curves from a scene, the description of a "cow wearing a dress" refines our visual understanding of relations on an object level and enables us to recognize a dressed cow when seeing it.

Relational reasoning is gaining growing popularity in the Computer Vision community and especially in the form of scene graph (SG) classification. The goal of SG classification is to classify objects and their relations in an image. One of the challenges in SG classification is collecting annotated image data. Most approaches in this domain rely on thousands of manually labeled and curated images. In this paper, we investigate whether the SG classification models can be fine-tuned from textual scene descriptions (similar to the "dressed cow" example above).

We consider a classification pipeline with two major parts: a feature extraction *backbone*, and a *relational reasoning* component (Figure 1). The backbone is typically a convolutional neural network (CNN) that detects objects and extracts an image-based representation for each. On the other hand, the relational reasoning component can be a variant of a recurrent neural network [Xu et al. 2017, Zellers et al. 2018] or graph convolutional networks [Yang et al. 2018, Sharifzadeh, Baharlou, and Tresp 2021]. This component operates on an object level by taking the latent representations of all the objects in the image and propagating them in the graph.

Note that, unlike the feature extraction backbone that requires images as input, the relational reasoning component operates on graphs with the nodes representing objects and the edges representing relations. The distinction between the input to the backbone (images) and the relational reasoning component (graphs) is often overlooked. Instead, the scene graph classification pipeline is treated as a network that takes only images as inputs. However, one can also train or fine-tune the relational reasoning component directly by injecting it with relational knowledge. For example, Knowledge Graphs (KGs) contain curated facts that indicate the relations between a `head` object and a `tail` object in the form of `(head, predicate, tail)` e.g., `(Person, Rides, Horse)`. The facts in KGs are represented by symbols whereas the inputs to the relational reasoning component are image-based embeddings. In this work, we map the triples to image-grounded embeddings as if they are coming from an image. We then use these embeddings to fine-tune the relational reasoning component through a denoising graph autoencoder scheme.

Note that the factual knowledge is not always available in a well-structured form, specially in domains where the knowledge is not stored in the machine-accessible form of KGs. In fact, most of the collective human knowledge is only

*These authors contributed equally.
†S. M. Baharlou contributed to this project while he was a visiting researcher at the Ludwig Maximilian University of Munich.

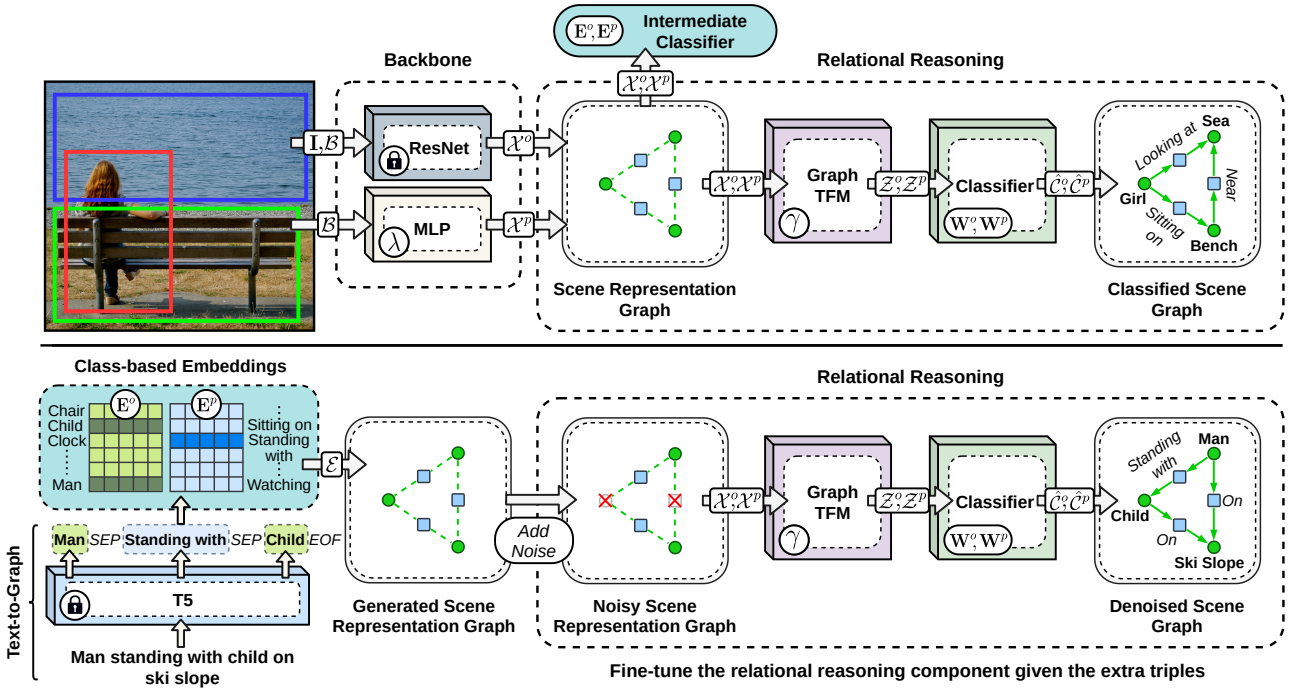

Figure 1: Top: we initially train a scene graph classification pipeline from images and their corresponding SGs. Bottom: we then use a text-to-graph module to extract structured knowledge from unstructured texts. The extracted graph is embedded by image-grounded vectors, masked, and then fed to the relational reasoning module to predict the missing relations and thus, encourage the network to learn the new relations from texts. The *lock* sign indicates pre-trained and frozen parts of the network.

available in the unstructured form of texts and documents. Exploiting this form of knowledge, in addition to structured knowledge, can be significantly beneficial. To this end, we employ a transformer-based model to generate structured graphs from textual input and utilize them to improve the relational reasoning module.

In summary, we propose *Texema*, a scene graph classification pipeline that can be trained from the large corpora of unstructured knowledge. We evaluate our approach on the Visual Genome dataset. In particular, we show that we can fine-tune the reasoning component using textual scene descriptions instead of thousands of images. As a result, when using as little as ∼500 images (1% of the VG training data), we can achieve ∼3x more accurate results in object classification, ∼8x in scene graph classification and ∼1.5x in predicate classification compared to the supervised baselines. Additionally, in our ablation studies, we evaluate the performance of using different rule-based, LSTM-based, and transformed-based text-to-graph models.

Related Works

Scene Graph Classification: There is an extensive body of work on visual reasoning in general that includes different forms of reasoning [Wu, Lenz, and Saxena 2014, Deng et al. 2014, Hu et al. 2016, 2017, Santoro et al. 2017, Zellers et al. 2019]. Here, we mainly review the works that are focused on scene graph classification. Visual Relation Detection (VRD) [Lu et al. 2016] and the Visual Genome [Krishna et al. 2017] are the main datasets for this task. While the original papers on VRD and VG provide the baselines for scene graph classification by treating objects independently, several follow-up works contextualize the entities before classification. Iterative Message Passing (IMP) [Xu et al. 2017], Neural Motifs [Zellers et al. 2018] (NM), Graph R-CNN [Yang et al. 2018], and Schemata [Sharifzadeh, Baharlou, and Tresp 2021] proposed to propagate the image context using basic RNNs, LSTMs, graph convolutions, and graph transformers respectively. On the other hand, authors of VTransE [Zhang et al. 2017] proposed to capture relations by applying TransE [Bordes et al. 2013], a knowledge graph embedding model, on the visual embeddings, Tang et al. [2019] exploited dynamic tree structures to place the object in an image into a visual context. Chen et al. [2019a] proposed a multi-agent policy gradient method that frames objects into cooperative agents and then directly maximizes a graph-level metric as the reward. In tangent to those works, Sharifzadeh et al. [2021] proposed to enrich the input domain in scene graph classification by employing the predicted pseudo depth maps of VG images that were released as an extension called *VG-Depth*.

Commonsense in Scene Understanding: Several recent works have proposed to employ external or internal sources of knowledge to improve visual understanding [Wang, Ye,

Input	man standing with child on ski slope
Reference Graph (RG)	**(child, on, ski slope) (man, on, ski slope) (man, standing with, child)**
$\text{R}_{\text{text}\rightarrow\text{graph}}$	*(man, standing, child)*
SSGP	*(standing, with, child) (standing, on, slope)*
CopyNet (1%)	**(man, standing with, child)**
T5 (1%)	**(man, standing with, child)**
CopyNet (10%)	**(man, standing with, child)** *(child, on, slope)*
T5 (10%)	**(man, standing with, child) (child, on, ski slope)**

Table 1: An example of extracted triples from a given text input in VG, using different methods. Bold: correct ($\in$ RG). Italic: incorrect ($\notin$ RG). The results are computed using the respective official code bases of the related works.

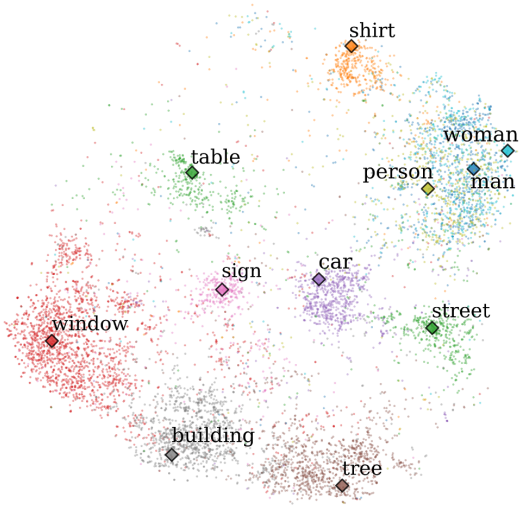

Figure 2: The t-SNE representation of the $\mathbf{e}_i$s (diamonds) and image-based representations $\mathcal{X}$s (dots) where each color represents the ground-truth class of the dot.

and Gupta 2018, Jiang et al. 2018, Singh et al. 2018, Kato, Li, and Gupta 2018]. In the scene graph classification domain, some of the works have proposed to correct the SG prediction errors by merely comparing them to the co-occurrence statistics of internal triples as a form of commonsense knowledge [Chen et al. 2019c,b, Zellers et al. 2018]. Earlier, Baier, Ma, and Tresp [2017, 2018] proposed the first scene graph classification model that employed prior knowledge in the form of Knowledge Graph Embeddings (KGEs) that generalize beyond the given co-occurrence statistics. Zareian, Karaman, and Chang [2020], Zareian et al. [2020] followed this approach by extending it to models that are based on graph convolutional networks. More recently, Sharifzadeh, Baharlou, and Tresp [2021] proposed Schemata as a generalized form of a KGE model that is learned directly from the images rather than triples. In general, scene graph classification methods are closely related to the KGE models. Therefore, we refer the interested readers to [Nickel et al. 2016, Ali et al. 2020a,b] for a review and large-scale study on the KG models, and to [Tresp, Sharifzadeh, and Konopatzki 2019, Tresp et al. 2020] for an extensive investigation of the connection between perception, KG models, and cognition.

Nevertheless, to the best of our knowledge, the described methods have employed curated knowledge in the form of triples, and none of them have directly exploited the textual knowledge. In this direction, the closest work to ours is by Yu et al. [2017], proposing to distill the external language knowledge using a teacher-student model. However, this work does not include a relational reasoning component and only refines the final predictions. Also, as shown in the experiments, our knowledge extraction module performs two times better than the SG Parser used in that work.

Knowledge Extraction from Text: Knowledge extraction from text has been studied for a long time [Chinchor 1991]. Previous work ranges from pattern-based approaches [Hearst 1992] to supervised neural approaches with specialized architectures [Gupta et al. 2019, Yaghoobzadeh, Adel, and Schütze 2017]. Recently, Schmitt et al. [2020] successfully applied a general sequence-to-sequence architecture to graph↔text conversion. With the recent rise of transfer learning in NLP, an increasing number of approaches are based on large language models, pre-trained in a self-supervised manner on massive amounts of texts [Devlin et al. 2019]. Inspired from previous work that explores transfer learning for graph-to-text conversion [Ribeiro et al. 2020], we base our text-to-graph model on a pre-trained T5 model [Raffel et al. 2019].

Methods

In this section, we first describe the backbone and relational reasoning components. We then describe our approach for fine-tuning the network from texts. We have three possible forms of data: Images (**IM**), Scene Graphs (**SG**) and Textual Scene Descriptions (**TXT**). We consider having two sets of data: one is the *parallel* set, which is the set of IM with their corresponding SG and TXT, and another is the *text* set which is a set of additional TXT that come without any images or scene graphs. These two sets have no elements in common.

We initially train our backbone and relational reasoning component from IM and SG, and our text-to-graph model from the TXT and SG in the parallel set. We then show that we can fine-tune the pipeline using the text set and without using any additional images.

Backbone (Algorithm 1.1)

The feature-extraction backbone is a convolutional neural network (ResNet-50) that has been pre-trained in a self-supervised manner [Grill et al. 2020] from unlabeled images of ImageNet [Deng et al. 2009] and Visual Genome [Krishna et al. 2017]. Given an image $\mathbf{I}$ with several objects in bounding boxes $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^n$, $\mathbf{b}_i = [b_i^x, b_i^y, b_i^w, b_i^h]$, we apply the ResNet-50 to extract pooled object features $\mathcal{X}^o = \{\mathbf{x}_i^o\}_{i=1}^n$, $\mathbf{x}_i^o \in \mathbb{R}^d$. Here $[b_i^x, b_i^y]$ are the coordinates of $\mathbf{b}_i$ and $[b_i^w, b_i^h]$ are its width and height, and d

Algorithm 1: Classify objects/predicates from images

1. **Extract image features (Backbone):**

   Input: Images and object bounding boxes ($\mathbf{I}, \mathcal{B} : \{\mathbf{b_i}\}_{i=1}^{n}$).

   Output: Object embeddings $\mathcal{X}^o : \{\mathbf{x}_i^o\}_{i=1}^{n}$ and predicate embeddings $\mathcal{X}^p : \{\mathbf{x}_i^p\}_{i=1}^{m}$.

   Trainable params: λ.

   $\mathcal{X}^o = ResNet50(\mathbf{I}, \mathcal{B})$

   $\mathcal{X}^p = \{MLP_\lambda(t(\mathbf{b}_i, \mathbf{b}_j)) \mid \forall \mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}\}$

2. **Contextualize and Classify (Relational Reasoning):**

   Input: Object embeddings $\mathcal{X}^o : \{\mathbf{x}_i^o\}_{i=1}^{n}$, Predicate embeddings $\mathcal{X}^p : \{\mathbf{x}_i^p\}_{i=1}^{m}$ and ground truth classes $\mathcal{C}^o$ and $\mathcal{C}^p$.

   Output: Predicted object class distribution $\hat{\mathcal{C}}^o : \{\hat{\mathbf{c}}_i^o\}_{i=1}^{n}$ and predicted predicate class distribution $\hat{\mathcal{C}}^p : \{\hat{\mathbf{c}}_i^p\}_{i=1}^{m}$.

   Trainable params: γ, $\mathbf{W}^o, \mathbf{W}^p$.

   $\mathcal{Z}^o, \mathcal{Z}^p = GraphTransformer_\gamma(\mathcal{X}^o, \mathcal{X}^p)$

   $\hat{\mathcal{C}}^o = \{\text{softmax}(\mathbf{W}^o \cdot \mathbf{z}^o) \mid \forall \mathbf{z}^o \in \mathcal{Z}^o\}$

   $\hat{\mathcal{C}}^p = \{\text{softmax}(\mathbf{W}^p \cdot \mathbf{z}^p) \mid \forall \mathbf{z}^p \in \mathcal{Z}^p\}$

3. **Apply Loss (Cross-Entropy):**

   $l_o = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\|\mathbf{c}_i^o\|} \mathbf{c}_{i,j}^o . log(\hat{\mathbf{c}}_{i,j}^o)$

   $l_p = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{\|\mathbf{c}_i^p\|} \mathbf{c}_{i,j}^p . log(\hat{\mathbf{c}}_{i,j}^p)$

are the vector dimensions. Following [Zellers et al. 2018], we define $\mathcal{X}^p = \{\mathbf{x}_i^p\}_{i=1}^{m}$, $\mathbf{x}_i^p \in \mathbb{R}^d$ as the relational features between each pair of objects. Each $\mathbf{x}_i^p$ is initialized by applying a two layered fully connected network on the relational position vector $\mathbf{t}$ between a head i and a tail j where $\mathbf{t} = [t_x, t_y, t_w, t_h]$, $t_x = (b_i^x - b_j^x)/b_i^w{}_j, t_y = (b_i^y - b_j^y)/b_j^h, t_w = \log(b_i^w/b_j^w), t_h = \log(b_i^h/b_j^h)$. The implementation and pre-training details of the layers are provided in the Evaluation. $\mathcal{X}^o$ and $\mathcal{X}^p$ form a structured presentation of the objects and predicates in the image also known as **Scene Representation Graph** (SRG) [Sharifzadeh, Baharlou, and Tresp 2021]. SRG is a fully connected graph with each node representing either an object or a predicate, where each object node is a direct neighbor to predicate nodes and each predicate node is a direct neighbor with its head and tail object nodes.

Relational Reasoning (Algorithm 1.2)

The relational reasoning component updates the initial SRG representations through Graph Transformer layers [Koncel-Kedziorski et al. 2019]. The outputs of these layers are $\mathcal{Z}^o = \{\mathbf{z}_i^o\}_{i=1}^{n}$, $\mathbf{z}_i^o \in \mathbb{R}^d$ and $\mathcal{Z}^p = \{\mathbf{z}_i^p\}_{i=1}^{m}$, $\mathbf{z}_i^p \in \mathbb{R}^d$ with equal dimensions as $\mathcal{X}$s. From here on, we drop the superscripts of o and p for brevity. We apply a linear classification layer $\mathbf{W}$ to classify the contextualized representations $\mathcal{Z}$ such that $\hat{\mathbf{c}} = \text{softmax}(\mathbf{W} \cdot \mathbf{z}_i)$, with cross-entropy as the loss function.

Fine-tuning from Texts (Algorithm 2)

Let us assume that we have already trained the backbone and relational reasoning components from IM and SG in the *parallel* set. Now, we want to fine-tune the weights in the relational reasoning component given the additional *text* set. The relational reasoning component takes graphs as input, therefore, we first need to convert TXT to SG:

Algorithm 2: Fine-tune the relational reasoning component from textual triples using a denoising auto-encoder paradigm

1. **Learn image-grounded representations E for each symbol through classification (without Graph Transformer):**

   Input: Object embeddings $\mathcal{X}^o : \{\mathbf{x}_i^o\}_{i=1}^{n}$, predicate embeddings $\mathcal{X}^p : \{\mathbf{x}_i^p\}_{i=1}^{m}$ and their corresponding ground truth classes $\mathcal{C}^o$ and $\mathcal{C}^p$.

   Output: Predicted object class distribution $\hat{\mathcal{C}}^o : \{\hat{\mathbf{c}}_i^o\}_{i=1}^{n}$ and predicted predicate class distribution $\hat{\mathcal{C}}^p : \{\hat{\mathbf{c}}_i^p\}_{i=1}^{m}$.

   Trainable params: $\mathbf{E}^o, \mathbf{E}^p$.

   $\hat{\mathcal{C}}^o = \{\text{softmax}(\mathbf{E}^o \cdot \mathbf{x}^o) \mid \forall \mathbf{x}^o \in \mathcal{X}^o\}$

   $\hat{\mathcal{C}}^p = \{\text{softmax}(\mathbf{E}^p \cdot \mathbf{x}^p) \mid \forall \mathbf{x}^p \in \mathcal{X}^p\}$

2. **Apply Loss (Cross Entropy):**

   $l_o = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\|\mathbf{c}_i^o\|} \mathbf{c}_{i,j}^o . log(\hat{\mathbf{c}}_{i,j}^o)$

   $l_p = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{\|\mathbf{c}_i^p\|} \mathbf{c}_{i,j}^p . log(\hat{\mathbf{c}}_{i,j}^p)$

3. **Fine-tune the relational reasoning component given the extra triples (Denoising Graph Autoencoder):**

   Input: Symbolic triples $\mathcal{S} : \{(h_i, p_i, t_i)\}_{i=1}^{k}$ and canonical object/predicate representations $\mathbf{E}^o/\mathbf{E}^p$.

   Output: Embedded representations $\mathcal{E} : \{(\mathbf{e}_i^h, \mathbf{e}_i^p, \mathbf{e}_i^t)\}_{i=1}^{k}$.

   Trainable params: $\gamma, \mathbf{W}^o, \mathbf{W}^p$.

   - Build $\mathcal{E} : \{(\mathbf{e}_i^h, \mathbf{e}_i^p, \mathbf{e}_i^t)\}_{i=1}^{k}$ where for each (h_i, p_i, t_i):
     $\mathbf{e}_i^h = onehot(h_i) \cdot \mathbf{E}^o$
     $\mathbf{e}_i^p = onehot(p_i) \cdot \mathbf{E}^p$
     $\mathbf{e}_i^t = onehot(t_i) \cdot \mathbf{E}^o$
   - Randomly set 20% of the nodes and edges in $\mathcal{E}$ to zero.
   - Set $\mathcal{X}^o = \mathcal{E}^h \cup \mathcal{E}^t$ and $\mathcal{X}^p = \mathcal{E}^p$ and run Algorithm 1.2 to fine-tune $\gamma, \mathbf{W}^o, \mathbf{W}^p$, with $\mathcal{E}^h$, $\mathcal{E}^t$ and $\mathcal{E}^p$ as the set of all heads, tails, and predicates in $\mathcal{E}$.

Text-to-graph: This model is trained from the SG and TXT in the parallel set, and then used to generate SG from the text set. Let us consider an unstructured text such as "man standing with child on ski slope" (Table 1 - Input). A structured form of this sentence is a graph with unique nodes and edges for each entity or predicate. For example, the reference graph for this sentence contains the triples `(child, on, ski slope)`, `(man, standing with, child)` and `(man, on, ski slope)` (Table 1 - RG).

In order to learn this mapping, we employ a transformer-based [Vaswani et al. 2017] sequence-to-sequence $T5_{small}$ model [Raffel et al. 2019] and adapt it for the task of extracting graphs from texts. T5 consists of an encoder with several layers of self-attention (like BERT, Devlin et al. 2019) and a decoder with autoregressive self-attention (like GPT-3, Brown et al. 2020). In order to use a T5 model with graphs, we need to represent the graphs as a sequence. To this end, we serialize the graphs by writing out their facts separated

Method	Precision		Recall		F1	
	1%	10%	1%	10%	1%	10%
$R_{text \rightarrow graph}$	1.92 ± 0.00	1.86 ± 0.01	1.87 ± 0.00	1.81 ± 0.01	1.89 ± 0.00	1.84 ± 0.01
SSGP	14.86 ± 0.01	14.52 ± 0.02	18.47 ± 0.01	18.05 ± 0.02	16.47 ± 0.01	16.09 ± 0.02
CopyNet	29.20 ± 0.13	30.77 ± 0.49	27.19 ± 0.28	29.79 ± 0.29	28.16 ± 0.21	30.27 ± 0.34
T5	$\mathbf{33.37 \pm 0.11}$	$\mathbf{33.81 \pm 0.08}$	$\mathbf{31.06 \pm 0.18}$	$\mathbf{32.45 \pm 0.33}$	$\mathbf{32.17 \pm 0.13}$	$\mathbf{33.12 \pm 0.16}$

Table 2: The mean and standard deviation of Precision, Recall, and F1 scores of the predicted facts from the texts on four random splits. The results are computed using the respective official code bases of the related works and evaluated on VG.

by end-of-fact symbols (EOF), and separate the elements of each fact with SEP symbols [Schmitt et al. 2020], e.g. *"child SEP on SEP ski slope EOF"* (Fig. 1). To adapt the multi-task setting from T5's pretraining, we use the task prefix "make graph: " to mark our text-to-graph task. Table 1 shows an example text and the extracted graphs using T5 and other previous methods (see Evaluation for details).

Map to embeddings: Note that the predicted graphs are a sequence of symbols for heads, predicates, and tails where each symbol represents a class $c \in \mathcal{C}$. However, the inputs to the relational reasoning component are image-based vectors $\mathcal{X}$. Thus, before feeding the symbols to the relational reasoning component, we need to map them to a corresponding embedding from the space of $\mathcal{X}$ as if we are feeding it with image-based embeddings. In order to do that, we train a mapping from symbols to $\mathcal{X}$s using the IM and SG of the parallel set. This is simply done by training a linear classification layer $\mathbf{E}$ given $\mathcal{X}$s from the parallel set (Algorithm 2.1). Unlike the classification layer in Algorithm 1, here we classify $\mathcal{X}$s instead of $\mathcal{Z}$s and the goal is *not* to use the classification output but to train image-grounded, canonical representations for each class: each row $\mathbf{e}_i$ in the classification layer becomes a cluster center for $\mathcal{X}$s from class i (Figure 2). Therefore, instead of the extracted symbolic c_i from the text set, we can feed its canonical image-grounded representation $\mathbf{e}_i$ to the graph transformer (Algorithm 2.3).

Denoising Graph Autoencoder: To fine-tune the relational reasoning given this data, we treat the relational reasoning component as a denoising autoencoder where the input is an incomplete (noisy) graph that comes from the text and the output is the denoised graph. If we do not apply a denoising autoencoder paradigm, the function will collapse to an identity map. We create the noisy graph by randomly setting some of the input nodes and edges to zero during the training (Algorithm 2.3). The goal is to encourage the graph transformer to predict the missing links and therefore, learn the relational structure.

Evaluation

We first compare the performance of different rule-based and embedding-based text-to-graphs models on our data. We then evaluate the performance of our entire pipeline in classifying objects and relations in images. In particular, we show that the extracted knowledge from the texts can largely substitute annotated images as well as ground-truth graphs.

Dataset: We use the sanitized version [Xu et al. 2017] of Visual Genome (VG) dataset [Krishna et al. 2017] including images and their annotations, i.e., bounding boxes, scene graphs, and scene descriptions. Our goal is to design an experiment that evaluates whether we can substitute annotated images with textual scene descriptions. Therefore, instead of using external textual datasets with unbounded information, we use Visual Genome itself by dividing it into different splits of *parallel* (with IM, SG and TXT) and *text* data (with only TXT). To this end, we assume only a random proportion (1% or 10%) of training images are annotated (parallel set containing IM with corresponding SG and TXT). We consider the remaining data (99% or 90%) as our text set and discard their IM and SG. We aim to see whether employing TXT from the text set, can substitute the discarded IM and SG from this set. We use four different random splits [Sharifzadeh, Baharlou, and Tresp 2021] to avoid a sampling bias. For more detail on the datasets refer to the supplementary materials.

Note that the scene graphs and the scene descriptions from the VG are collected separately and by crowd-sourcing. Therefore, even though the graphs and the scene descriptions refer to the same image region, they are disjoint and contain complementary knowledge.

Graphs from Texts

The goal of this experiment is to study the effectiveness of the text-to-graph model. We fine-tune the pre-trained T5 model on parallel TXT and SG, and apply it on the text set to predict their corresponding SG. We also implement the following rule-based and embedding-based baselines to compare their performance using our splits: (1) $R_{text \rightarrow graph}$ is a simple rule-based system introduced by Schmitt et al. [2020] for general knowledge graph generation from text. (2) The Stanford Scene Graph Parser (SSGP) [Schuster et al. 2015] is another rule-based approach that is more adapted to the scene graph domain. Even though this approach was not specifically designed to match the scene graphs from the Visual Genome dataset, it was still engineered to cover typical idiosyncrasies of textual image descriptions and corresponding scene graphs. (3) CopyNet [Gu et al. 2016] is an LSTM sequence-to-sequence model with a dedicated copy mechanism, which allows copying text elements directly into the graph output sequence. It was used for unsupervised text-to-graph generation by Schmitt et al. [2020]. However, we train it on the supervised data of our parallel sets. We use a vocabulary of around 70k tokens extracted from the VG-graph-text

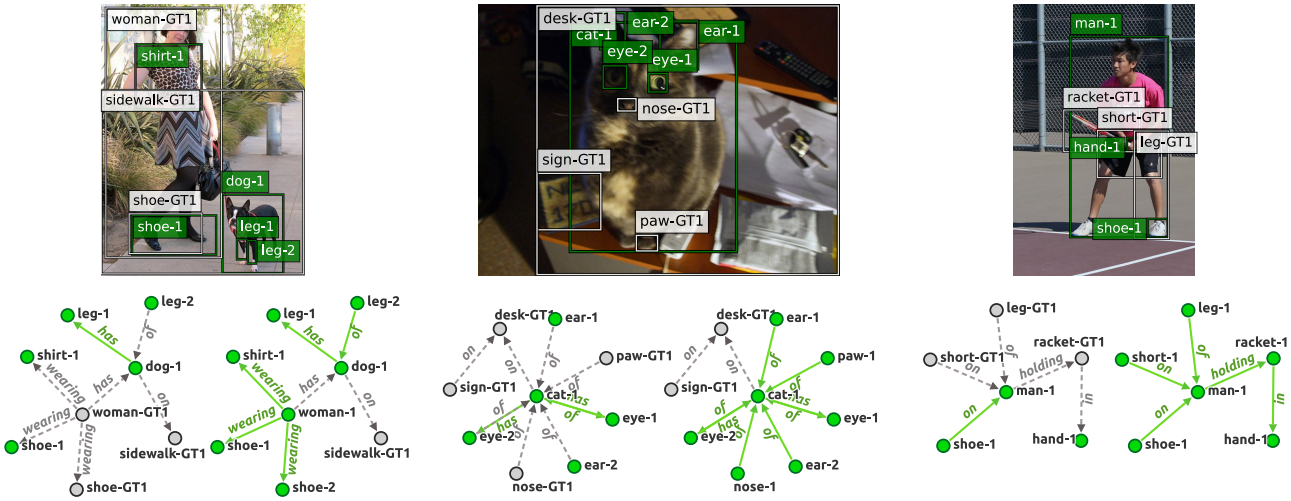

Figure 3: Qualitative examples of improved classification results (Recall@100) before and after (from left to right) fine-tuning the model using the knowledge in texts. Green and gray colors indicate true positives and false negatives concluded by the model.

	Method	R@50		R@100	
		1%	10%	1%	10%
SGCls	$R_{text \to graph}$	10.90 ± 0.12	24.96 ± 0.15	11.80 ± 0.11	26.09 ± 0.15
	SSGP	14.35 ± 0.15	26.11 ± 0.19	15.14 ± 0.17	27.12 ± 0.22
	CopyNet	14.46 ± 0.31	26.05 ± 0.29	15.19 ± 0.24	27.08 ± 0.26
	TXM - T5	$\mathbf{14.53 \pm 0.34}$	$\mathbf{26.16 \pm 0.32}$	$\mathbf{15.28 \pm 0.38}$	$\mathbf{27.22 \pm 0.28}$
	GT	14.72 ± 0.38	26.33 ± 0.45	15.36 ± 0.38	27.37 ± 0.47
PredCls	$R_{text \to graph}$	23.34 ± 0.10	49.99 ± 0.12	26.83 ± 0.15	54.40 ± 0.12
	SSGP	54.65 ± 0.14	55.65 ± 0.15	59.33 ± 0.18	59.67 ± 0.20
	CopyNet	56.24 ± 0.31	59.27 ± 0.28	60.35 ± 0.20	63.28 ± 0.25
	TXM - T5	$\mathbf{58.64 \pm 0.34}$	$\mathbf{59.31 \pm 0.30}$	$\mathbf{63.07 \pm 0.37}$	$\mathbf{63.32 \pm 0.24}$
	GT	62.02 ± 0.10	61.71 ± 0.19	65.68 ± 0.12	65.42 ± 0.19

Table 3: SGCls and PredCls results using different text-to-graph modules. We have substituted the missing 99% and 90% of annotated images with the textual knowledge extracted from their scene descriptions.

benchmark and, otherwise, also adopt the hyperparameters from [Schmitt et al. 2020]. Table 1 shows sample predictions from these models. Table 2 compares precision, recall, and F1 measures. and T5 outperforms others by a large margin.

Graphs from Images

The goal of this experiment is to evaluate scene graph classification after fine-tuning the pipeline using textual knowledge. We evaluate our models for object classification, predicate classification (PredCls - predicting predicate labels given a ground truth set of object boxes and object labels) and scene graph classification (SGCls - predicting object and predicate labels, given the set of object boxes) on the test sets. Our ablation study concerns the following configurations:

- **SPB**: In this setting, both the backbone and the relational reasoning component are trained by *supervised learning* on the IM and SGs (1% or 10%) from the parallel set.
- **SCH**: Here, the backbone is trained by *self-supervised learning* on all VG images (without labels), and the relational reasoning component is trained on the IM and SGs (1% or 10%) from the parallel set.
- **TXM**: Here, the backbone is trained by *self-supervised learning* on all VG images (without labels), and the relational reasoning component is trained on the IM and SGs (1% or 10%) from the parallel set and fine-tuned from the SGs predicted from the text set (99% or 90%) using the text-to-graph module.
- **GT**: Here, the backbone is trained by *self-supervised learning* on all VG images (without labels), and the relational reasoning component is trained on the IM and SGs (1% or 10%) from the parallel set, and fine-tuned from the *ground truth graphs* (99% or 90%), instead of the text-to-graph predictions.

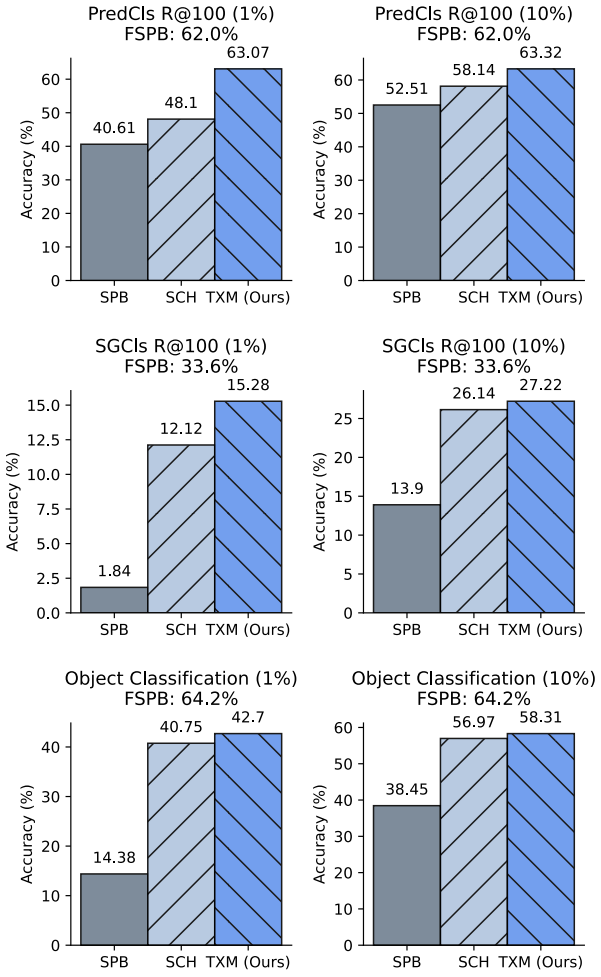

Figure 4: Fine-tuning with the textual knowledge (TXM) significantly improves the results in all settings of PredCls (top), SGCls (middle), and object classification (bottom).

- **FSPB**: Here, both the backbone and the relational reasoning component are trained by *supervised learning* on 100% of the VG annotated images. Meaning that we have redefined the parallel set to include 100% of the VG training data and we do not need to substitute the images with the text set anymore. The goal of this setting is to compute the maximum accuracy that our model achieves, when we have all the annotated images with ground truth SGs, instead of using their textual scene descriptions. The results of this settings are written above each table so that the other bars maintain a meaningful scale.

Figure 4 presents the results of the ablation study. We use the Recall@K (**R@K**) as metric. which computes the mean prediction accuracy in each image given the top K predictions. For more results (mR@K metric and unconstrained setups) refer to the supplementary materials. As shown, fine-tuning with textual scene descriptions (TXM) improves the classification results under all settings, substituting a large proportion of the omitted images. Furthermore, the results even outperform FSPB under PredCls (recall that the scene descriptions are sometimes complementary to image annotations and contain additional information).

Method	SGCls		PredCls	
	R@50	R@100	R@50	R@100
VRD [Lu et al. 2016]	11.8	14.1	27.9	35.0
IMP+ [Xu et al. 2017]	34.6	35.4	59.3	61.3
SMN [Zellers et al. 2018]	35.8	36.5	65.2	67.1
KERN [Chen et al. 2019c]	36.7	37.4	65.8	67.6
VCTree [Tang et al. 2019]	38.1	38.8	66.4	68.1
CMAT [Chen et al. 2019a]	39.0	39.8	66.4	68.1
SIG [Wang et al. 2020]	36.6	37.3	66.3	68.1
GB-Net [Zareian et al. 2020]	38.0	38.8	66.6	68.2
TXM	**39.0**	**39.9**	**66.7**	**68.3**

Table 4: Comparing the general performance of the architecture to some other methods under the VG test set.

Table 3 presents additional results also using different text-to-graph baselines. We can see that fine-tuning with the predicted graphs using T5, is as effective as fine-tuning with the crowd-sourced ground truth graphs (GT), and in some settings even better (object classification with 1%). Notice that compared to the self-supervised baseline, we gained up to $\sim$5% *relative improvement* in object classification, more than $\sim$26% in scene graph classification, and $\sim$31% in predicate prediction accuracy. As expected, the choice of text-to-graph module has a larger effect on the PredCls compared to the SGCls and ObjCls, due to the fact that SGCls and ObjCls rely heavily on the image-based features, whereas PredCls has a strong dependency to relational knowledge. In supplementary materials we also provide additional results on the improvements per object class after fine-tuning the model with the textual knowledge (From SCH to TXM) and show that most improvements occur in under-represented classes. Figure 3 provides some qualitative examples of the predicted scene graphs before and after fine-tuning with the texts. Finally, to provide an intuition on our general performance, Table 4 present the results of our architecture using a VGG-16 [Simonyan and Zisserman 2014] backbone trained with 100% of the annotations, instead of the self-supervised BYOL.

Conclusion

In this work, we proposed the first relational image-based classification pipeline that can be fine-tuned directly from the large corpora of unstructured knowledge available in texts. We generated structured graphs from textual input using different rule-based or embedding-based approaches. We then fine-tuned the relational reasoning component of our classification pipeline by employing the canonical representations of each entity in the generated graphs. We showed that we gain a significant improvement in all settings after employing the inferred knowledge within the classification pipeline. In most cases, the accuracy was similar to when using the ground truth graphs that are manually annotated by crowd-sourcing.

Acknowledgments

We would like to thank Masoud Jalili Sabet for the fruitful discussions, and the anonymous reviewers for their helpful feedback on the manuscript. This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A.

References

Ali, M.; Berrendorf, M.; Hoyt, C. T.; Vermue, L.; Galkin, M.; Sharifzadeh, S.; Fischer, A.; Tresp, V.; and Lehmann, J. 2020a. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *arXiv preprint arXiv:2006.13365*.

Ali, M.; Berrendorf, M.; Hoyt, C. T.; Vermue, L.; Sharifzadeh, S.; Tresp, V.; and Lehmann, J. 2020b. Pykeen 1.0: A python library for training and evaluating knowledge graph emebddings. *arXiv preprint arXiv:2007.14175*.

Baier, S.; Ma, Y.; and Tresp, V. 2017. Improving visual relationship detection using semantic modeling of scene descriptions. In *International Semantic Web Conference*, 53–68. Springer.

Baier, S.; Ma, Y.; and Tresp, V. 2018. Improving information extraction from images with learned semantic models. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 5214–5218. AAAI Press.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *Computing Research Repository*, arXiv:2005.14165.

Chen, L.; Zhang, H.; Xiao, J.; He, X.; Pu, S.; and Chang, S.-F. 2019a. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, 4613–4623.

Chen, T.; Xu, M.; Hui, X.; Wu, H.; and Lin, L. 2019b. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 522–531.

Chen, T.; Yu, W.; Chen, R.; and Lin, L. 2019c. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6163–6171.

Chinchor, N. 1991. MUC-3 Linguistic Phenomena Test Experiment. In *Third Message Uunderstanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*.

Deng, J.; Ding, N.; Jia, Y.; Frome, A.; Murphy, K.; Bengio, S.; Li, Y.; Neven, H.; and Adam, H. 2014. Large-scale object classification using label relation graphs. In *European conference on computer vision*, 48–64. Springer.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1631–1640. Berlin, Germany: Association for Computational Linguistics.

Gupta, P.; Rajaram, S.; Schütze, H.; and Runkler, T. A. 2019. Neural Relation Extraction within and across Sentence Boundaries. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 6513–6520.

Hearst, M. A. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Hu, H.; Deng, Z.; Zhou, G.-T.; Sha, F.; and Mori, G. 2017. Labelbank: Revisiting global perspectives for semantic segmentation. *arXiv preprint arXiv:1703.09891*.

Hu, H.; Zhou, G.-T.; Deng, Z.; Liao, Z.; and Mori, G. 2016. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2960–2968.

Jiang, C.; Xu, H.; Liang, X.; and Lin, L. 2018. Hybrid knowledge routed modules for large-scale object detection. *arXiv preprint arXiv:1810.12681*.

Kato, K.; Li, Y.; and Gupta, A. 2018. Compositional learning for human object interaction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 234–251.

Koncel-Kedziorski, R.; Bekal, D.; Luan, Y.; Lapata, M.; and Hajishirzi, H. 2019. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1904.02342*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al.

2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1): 32–73.

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2016. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 852–869. Springer.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1): 11–33.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Ribeiro, L. F. R.; Schmitt, M.; Schütze, H.; and Gurevych, I. 2020. Investigating Pretrained Language Models for Graph-to-Text Generation. *Computing Research Repository*, arXiv:2007.08426.

Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, 4967–4976.

Schmitt, M.; Sharifzadeh, S.; Tresp, V.; and Schütze, H. 2020. An unsupervised joint system for text generation from knowledge graphs and semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7117–7130.

Schuster, S.; Krishna, R.; Chang, A.; Fei-Fei, L.; and Manning, C. D. 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, 70–80.

Sharifzadeh, S.; Baharlou, S. M.; Berrendorf, M.; Koner, R.; and Tresp, V. 2021. Improving Visual Relation Detection using Depth Maps. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 3597–3604.

Sharifzadeh, S.; Baharlou, S. M.; and Tresp, V. 2021. Classification by Attention: Scene Graph Classification with Prior Knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 5025–5033.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, K. K.; Divvala, S.; Farhadi, A.; and Lee, Y. J. 2018. Dock: Detecting objects by transferring common-sense knowledge. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 492–508.

Tang, K.; Zhang, H.; Wu, B.; Luo, W.; and Liu, W. 2019. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6619–6628.

Tresp, V.; Sharifzadeh, S.; and Konopatzki, D. 2019. A Model for Perception and Memory. *Conference on Cognitive Computational Neuroscience*.

Tresp, V.; Sharifzadeh, S.; Konopatzki, D.; and Ma, Y. 2020. The Tensor Brain: Semantic Decoding for Perception and Memory. *arXiv preprint arXiv:2001.11027*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, W.; Wang, R.; Shan, S.; and Chen, X. 2020. Sketching image gist: Human-mimetic hierarchical scene graph generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, 222–239. Springer.

Wang, X.; Ye, Y.; and Gupta, A. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6857–6866.

Wu, C.; Lenz, I.; and Saxena, A. 2014. Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception. In *Robotics: Science and systems*.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5419.

Yaghoobzadeh, Y.; Adel, H.; and Schütze, H. 2017. Noise Mitigation for Neural Entity Typing and Relation Extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1183–1194. Valencia, Spain: Association for Computational Linguistics.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 670–685.

Yu, R.; Li, A.; Morariu, V. I.; and Davis, L. S. 2017. Visual relationship detection with internal and external linguistic knowledge distillation. In *IEEE International Conference on Computer Vision (ICCV)*.

Zareian, A.; Karaman, S.; and Chang, S.-F. 2020. Bridging knowledge graphs to generate scene graphs. In *European Conference on Computer Vision*, 606–623. Springer.

Zareian, A.; You, H.; Wang, Z.; and Chang, S.-F. 2020. Learning Visual Commonsense for Robust Scene Graph Generation. *arXiv preprint arXiv:2006.09623*.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6720–6731.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5831–5840.

Zhang, H.; Kyaw, Z.; Chang, S.; and Chua, T. 2017. Visual Translation Embedding Network for Visual Relation Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 3107–3115. IEEE Computer Society. ISBN 978-1-5386-0457-1.

4 Improving visual relation detection using depth maps

This chapter comprises the publication

Sharifzadeh et al. [2020]

and the code is available at

`https://github.com/Sina-Baharlou/Depth-VRD`

Declaration of Authorship The research idea was proposed by Sahand Sharifzadeh and discussed with Max Berrendorf. Sahand Sharifzadeh did the main implementation and conducted experiments; Sina Moayed Baharlou contributed to parts of the code, Max Berrendorf and Rajat Koner contributed smaller parts of the code. The final manuscript was mainly written by Sahand Sharifzadeh and revised by all authors.

- *In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of LMU's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to* `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` *to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.*

Improving Visual Relation Detection using Depth Maps

Sahand Sharifzadeh
Ludwig Maximilian University of Munich
sharifzadeh@dbs.ifi.lmu.de

Sina Moayed Baharlou
Sapienza University of Rome
baharlou@dis.uniroma1.it

Max Berrendorf
Ludwig Maximilian University of Munich
berrendorf@dbs.ifi.lmu.de

Rajat Koner
Ludwig Maximilian University of Munich
koner@dbs.ifi.lmu.de

Volker Tresp
Ludwig Maximilian University of Munich
& Siemens AG
volker.tresp@siemens.com

***Abstract*—Visual relation detection methods rely on object information extracted from RGB images such as 2D bounding boxes, feature maps, and predicted class probabilities. We argue that depth maps can additionally provide valuable information on object relations, e.g. helping to detect not only spatial relations, such as `standing behind`, but also non-spatial relations, such as `holding`. In this work, we study the effect of using different object features with a focus on depth maps. To enable this study, we release a new synthetic dataset of depth maps, *VG-Depth*, as an extension to Visual Genome (VG). We also note that given the highly imbalanced distribution of relations in VG, typical evaluation metrics for visual relation detection cannot reveal improvements of under-represented relations. To address this problem, we propose using an additional metric, calling it *Macro Recall@K*, and demonstrate its remarkable performance on VG. Finally, our experiments confirm that by effective utilization of depth maps within a simple, yet competitive framework, the performance of visual relation detection can be improved by a margin of up to 8%.**

***Index Terms*—scene graph, visual relation detection, depth maps**

I. Introduction

Scene Graph Generation, i.e. detecting objects and their relations in images in form of `(subject, predicate, object)`, is a fundamental task in scene understanding and can play an important role in recommender systems, visual question answering, decision making, etc. For example, detecting whether a man is `on` a bike or `next to` a bike is a crucial challenge in autonomous driving. Most works in this area rely on image-based object information such as class labels, bounding boxes and RGB features. We argue that depth maps can additionally provide valuable information about an object's relations as they provide the objects' distance from the camera. This information can help to distinguish between many relations such as `behind`, `in front of` and even improve detection in situations where the objects are nearby such as `covered in`. Figure 1 shows a successfully detected example of the relation `(fence, behind, dog)` after employing its depth map, and using our model. The goal of this work is to study the effect of using different object features on visual relation detection, with a focus on depth maps.

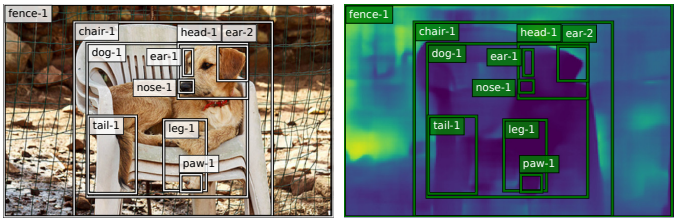

Fig. 1. An image from the VG dataset (left), and the corresponding synthetically generated depth map from VG-Depth dataset (right), annotated by the scene graph. Bright colors in the depth map indicate a larger distance to the camera. Utilizing depth maps allows us to successfully predict the relation `(fence-1, behind, dog-1)`.(© 2020 IEEE)

Unfortunately, most available image datasets, specifically the ones with relational annotations such as Visual Relation Detection (VRD) [2] and Visual Genome (VG) [3], do not provide depth maps, because the acquisition of depth maps is a cumbersome task requiring specialized hardware. We tackle this issue by synthetically generating the corresponding *pseudo* depth maps from 2D images of Visual Genome. This is possible thanks to the large corpora of available RGB-D pairs, i.e. NYU-Depth-v2 [4] dataset. Using RGB-D pairs in NYU-Depth-v2 and a fully convolutional neural network, allow us to learn the mapping function of RGB images to their corresponding depth maps. We can then apply this network to the images from VG, generating their corresponding depth maps. We release the depth maps that are generated from VG, as an extention to it, calling it ***VG-Depth***[1]. The object information extracted from depth maps and RGB images, i.e. class labels, location vectors, RGB and depth features, are the basis for relation detection in our simple yet effected framework.

Additionally, we note that the typically employed Recall@K metric (Micro Recall@K), cannot properly reveal the improvements of under-represented relations in highly imbalanced datasets such as VG. This might be an issue in applications such as autonomous driving where it is important to ensure

[1]The dataset and our framework are publicly available at https://github.com/Sina-Baharlou/Depth-VRD.

that the model is capable of predicting also important but less represented predicates such as *walking on* (648 in VG test set) and not just *wearing* (20,148 in VG test set). We address this issue by proposing to employ ***Macro Recall@K***, where we compute the mean over Micro Recall@K per predicate, thereby eliminating the effect that over-represented classes have in Micro Recall@K setting.

In summary, our contributions are as follows:

1) We perform an extensive study on the effect of using different sources of object information in visual relation detection. We show in our empirical evaluations using the VG dataset, that our model can outperform competing methods by a margin of up to 8% points, even those using external language sources or contextualization.
2) We release a new synthetic dataset *VG-Depth*, to compensate for the lack of depth maps in Visual Genome.
3) We propose *Macro Recall@K* as a competitive metric for evaluating the visual relation detection performance in highly imbalanced datasets such as Visual Genome.

II. Related Works

a) Knowledge Graph (KG) Modeling: In Knowledge Graph modeling, the aim is typically to find embeddings or latent representations for entities and predicates, which then can serve to predict the probability of unseen triples. These methods mostly differ in how they model relations. In RESCAL [5] each relation is defined as a transformation in the embedding space of entities, producing a triple probability. TransE [6] employs a similar idea but limits each relation to a translation. In comparison to RESCAL, it has fewer parameters; as a disadvantage, it cannot model symmetric relations. DistMult [7] considers each relation as a vector, similar to TransE, but minimizes the trilinear dot product of subject, predicate and object vector. DistMult can be understood as a form of RESCAL, where the transformation matrix is diagonal. ComplEx [8] extends DistMult to complex-valued vectors of embeddings. A multilayer perceptron (MLP) architecture [9] extends these methods to non-linear transformations and has shown to be competitive to the other discussed approaches on most benchmarks [10], [11]. For an extensive review and study on different KG models refer to [10], [12], [13].

b) Scene Graph (SG) Generation: SG Generation started with the release of Visual Relation Detection (VRD) [2] and the VG [3]. In VRD, Word2Vec representations of the subject, object, and the predicate were used to train a model jointly with the corresponding image region that describes the predicate. In particular, they consider the joint bounding box of subject and object as the image representation for the predicate. Follow-up work achieved improved performance by incorporating a knowledge graph, constructed from the image annotations [14]. Later, VTransE employed TransE [15] to model visual relations. More recently, Yu et al. [16] proposed a teacher-student model to distill external language knowledge to improve visual relation detection. Iterative Message Passing [17], Neural Motifs [18] (NM) and Graph R-CNN [19] incorporate context within each prediction using RNNs and graph convolutions respectively. For an extensive discussion on the connection between scene graphs and knowledge graphs refer to [20], [21].

c) Depth Maps: While several works have leveraged depth maps to improve *object* detection [22]–[24], the idea of using depth maps in the *relation* detection task has only been explored recently: Yang et al. [25] employ a basic framework for visual relation detection, with handcrafted depth map features, i.e. the mean and mode over pixel values of each depth map. They have a limited experimental setting, where they consider only human-centered relations. In this work, we explore the usability of depth maps in a larger domain and using a convolutional neural network for feature extraction. Furthermore, we provide a more extensive study, release a relevant dataset, and propose a more suitable metric.

III. Framework

In this section, we introduce the framework that we employed for this study. Let $\mathcal{E} = \{e_1, e_2, ..., e_n\}$ be the set of all entities, including subjects (s) and objects (o), and $\mathcal{P} = \{p_1, p_2, ..., p_m\}$ the set of all predicates. Each entity e_i can appear in images within a bounding box $\mathbf{bb_i} = [x_i, y_i, w_i, h_i]$, from an image $\mathbf{I}$, where $[x_i, y_i]$ are the coordinates of the bounding box and $[w_i, h_i]$ are its width and height. In this work we apply Faster R-CNN [26], on each image $\mathbf{I}$ to extract a feature map $\mathbf{fmap_I}$, together with object proposals as a set of bounding boxes $\mathbf{bb}$ and class probability distributions $\mathbf{c}$. For each RGB image, we generate a depth map $\mathbf{D}$ where the same bounding box areas encompass the entities' distance from the camera. In the next section, we first describe the synthetic generation of $\mathbf{D}$s and then the feature extraction from generated depth maps. In the end, we describe the relation detection module, where the pairwise features are fused and then employed for relation detection.

A. Depth Maps for Relation Detection

1) Generation: We incorporate an RGB-to-Depth model within our visual relation detection framework. As shown in Figure 2, this is a fully convolutional neural network (CNN) that takes an RGB image as input and generates its predicted depth map. This model can be pre-trained on any datasets containing pairs of RGB and depth maps regardless of having the class annotations for objects or predicates. This enables us to work with the already available visual relation detection datasets without requiring to collect additional data, and also mitigates the need for specialized hardware in real-world applications. The architectural details are explained in Section IV and the generated depth maps from VG are separately released as a dataset called *VG-Depth*.

2) Feature Extraction: Depth maps have been employed in tasks such as *object detection* and *segmentation* [23], [27]. In these works, it is common to simply render a depth map as an RGB image, and extract depth features using a CNN, that has been pre-trained with RGB images (for object detection). They argue that the edges in depth maps might

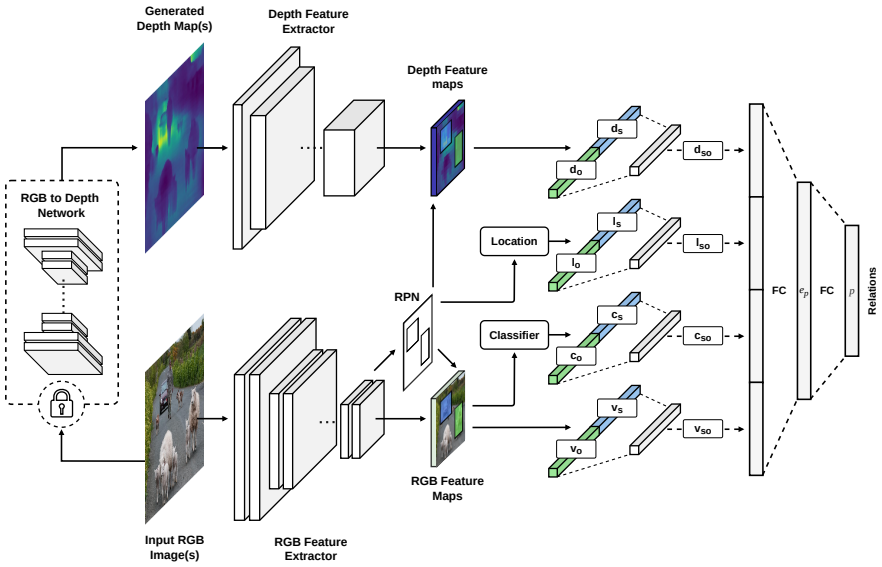

Fig. 2. We study the effect of object information, i.e. class labels, location vectors, RGB and depth features in visual relation detection by employing the simple yet effective framework presented in this figure. We generate depth maps synthetically using an RGB-to-Depth model, eliminating the need for specialized hardware. On the left side, we see the RGB image and its generated depth map, fed into CNNs to extract feature maps from both modalities. We create pairwise feature vectors $\mathbf{d_{so}}$ (pooled from depth feature maps), $\mathbf{l_{so}}$ (from bounding boxes), $\mathbf{c_{so}}$ (from class labels) and $\mathbf{v_{so}}$ (pooled from RGB features) and feed them into a relation detection layer to infer the predicate (© 2020 IEEE).

yield better object contours than the edges in cluttered RGB images and that one may combine edges from both RGB and depth to obtain more information [27]. Therefore, they aim to get similar, complementary features from both modalities. However, the practice of employing a model pre-trained on a particular source modality, e.g. RGB, and applying it on a different target modality, e.g. depth map, is sub-optimal in many applications (one should also keep in mind that even fine-tuning some layers of a network does not change the very early convolutional filters). Hence, unlike other works, we train a feature extractor CNN directly on depth maps and specifically for the task of relation detection. Given a depth map $\mathbf{D}$, this network generates a feature map $\mathbf{fmap_D}$. The architectural details of this network is presented in Section IV.

B. Relation Model

In the previous section, we described methods for the extraction of $\mathbf{fmap_I}$, $\mathbf{fmap_D}$, $\mathbf{c}$ and $\mathbf{bb}$. Here, we outline the model that infers relations using pairwise combinations of these features. For each pair of detected objects within an image, we create a scale-invariant location feature $\mathbf{l_s} = [t_x, t_y, t_w, t_h]$ with: $t_x = (x_s - x_o)/w_o, t_y = (y_s - y_o)/h_o, t_w = \log(w_s/w_o), t_h = \log(h_s/h_o)$ and similarly $\mathbf{l_o}$. We then pool the corresponding features $\mathbf{v_s}$ and $\mathbf{v_o}$ from $\mathbf{fmap_I}$ and create a visual feature vector $[\mathbf{v_s}; \mathbf{v_o}]$. Similarly, we create a depth feature vector $[\mathbf{d_s}; \mathbf{d_o}]$, by pooling features from $\mathbf{fmap_D}$, within $\mathbf{bb}_s$ and $\mathbf{bb}_o$. Additionally, we create $[\mathbf{c_s}; \mathbf{c_o}]$ and $[\mathbf{l_s}; \mathbf{l_o}]$. Each of these vectors are fed into separate fully connected layers, followed by ReLUs, yielding $\mathbf{v_{so}}$, $\mathbf{l_{so}}$, $\mathbf{c_{so}}$ and $\mathbf{d_{so}}$ before being fed to the relation head which projects them to the relation space such that:

$$\mathbf{e_p} = f(\mathbf{W}[\mathbf{v_{so}}; \mathbf{l_{so}}; \mathbf{c_{so}}; \mathbf{d_{so}}]) \quad (1)$$

Here, $\mathbf{W}$ describes a linear transformation and $f(.)$ is a non-linear function. We realize them as a fully connected layer in a neural network with ReLU activations and dropout. $\mathbf{e_p}$ is an embedding vector of pairwise features. This simple relation prediction model is inspired by the work of [9] to predict links in knowledge graphs. Therefore, we call it **ERMLP-E**, short for ERMLP-Extended. The input of their proposed model is a triple and the output is a single Bernoulli variable, whereas in our work the inputs are *subject* and *object* and we have a Bernoulli variable for each predicate class in the output. This gives us fewer parameters compared to that model, and simplifies training by imposing an implicit negative sampling through the cross-entropy loss.

As shown in earlier works, using more sophisticated models for context propagation between objects with RNNs or graph convolutions, can further improve the prediction accuracy. However, the aim here is to study the effect of including depth maps as additional object features in visual relation detection and as will be shown later, even with this simple model, utilizing depth maps can be more effective than e.g. propagating context. Clearly, those other models can also further enrich their understanding of object relations by employing depth maps.

To learn the parameters, we consider each relation `(subject, predicate, object)` with an associated Bernoulli variable that takes 1 if the triple is observed and 0 otherwise, following a locally closed world assumption [10].

TABLE I

PREDICATE PREDICTION RECALL VALUES ON VG TEST SET. WHEN THE DEPTH MAPS ARE UTILIZED TOGETHER WITH ALL OTHER FEATURES (*Ours*-$\mathbf{l}, \mathbf{c}, \mathbf{v}, \mathbf{d}$), WE GAIN A LARGE IMPROVEMENT COMPARED TO THE STATE-OF-THE-ART. ONE CAN ALSO SEE THAT EVEN REPLACING DEPTH MAPS WITH VISUAL FEATURES (*Ours*-$\mathbf{l}, \mathbf{c}, \mathbf{d}$ COMPARED TO *Ours*-$\mathbf{l}, \mathbf{c}, \mathbf{v}$) CAN YIELD BETTER RESULTS. ADDITIONALLY, COMPARING *Ours*-$\mathbf{l}, \mathbf{c}, \mathbf{v}$ TO *VTransE* AND *Neural Motifs* REVEALS THE ADVANTAGE OF OUR SIMPLE MODEL REGARDLESS OF DEPTH MAPS. (© 2020 IEEE)

	Strategy	**Macro**			**Micro**		
	Task	Predicate Pred.			Predicate Pred.		
	Metric	R@100	R@50	R@20	R@100	R@50	R@20
models	VTransE [28]	-	-	-	62.87	62.63	-
	Yu's-S [16]	-	-	-	49.88	-	-
	Yu's-S+T [16]	-	-	-	55.89	-	-
	IMP [17]	-	-	-	53.00	44.80	-
	Graph R-CNN [19]	-	-	-	59.10	54.20	-
	NM [18]	14.39	13.20	10.25	67.10	65.20	58.50
ablations	Ours - d	9.51	8.46	6.35	54.72	51.90	43.86
	Ours - c	15.65	13.09	8.56	64.82	60.54	49.89
	Ours - v	13.88	12.24	8.99	61.72	58.50	50.41
	Ours - l	5.19	4.66	3.57	49.07	46.13	37.48
	Ours - v, d	15.47	14.04	10.83	62.88	60.52	53.07
	Ours - l, v, d	15.76	14.40	11.07	63.06	60.83	53.55
	Ours - l, c, d	21.67	19.56	15.12	67.97	66.09	59.13
	Ours - l, c, v	19.16	17.72	13.93	67.94	66.06	59.14
	Ours - l, c, v, d	**22.72**	**20.74**	**16.40**	**68.00**	**66.18**	**59.44**

Given the set of observed triples $\mathcal{T}$, the loss function is the categorical cross entropy between the one-hot targets and the distribution obtained by softmax over the network's output defined as:

$$\mathcal{L} = \sum_{(s,p,o)\in\mathcal{T}} -\log \frac{\exp\left(\mathbf{w'}_p^{\mathrm{T}}\mathbf{e_p}\right)}{\sum_{p'\in\mathcal{P}}\exp\left(\mathbf{w'}_{p'}^{\mathrm{T}}\mathbf{e_p}\right)} \tag{2}$$

where $\mathbf{w'}_p$ is the weight vector corresponding to p in the last layer (linear classification layer).

IV. EVALUATION

In our study, we are interested to answer the following questions:

1) If we are given *only* depth maps of some objects in a scene (and not even object labels), how accurately can we infer the distribution of possible pairwise relations? How do other sources of object information compare to it?
2) Current visual relation detection frameworks commonly rely on extensive object information such as class labels, bounding boxes, RGB features, contextual information, etc. Do depth representations bring any additional information or would they only contribute redundant scene knowledge?

Additionally, we study whether Recall@K can sufficiently reflect the improvements of under-represented relations within a highly imbalanced dataset such as VG.

In what follows, we introduce the dataset, metrics, architectural details and experiments to answer these questions.

A. Dataset

We test our approach on the *Visual Genome* [3] dataset. We use the more commonly used subset of VG dataset proposed by [17] which contains 150 object classes and 50 relations.

B. Metrics

a) Micro Recall@K: This metric is defined as the mean prediction accuracy in each image given the top K predictions and is typically called *Recall@K*. We assigned the *Micro* prefix to its name to distinguish this metric with *Macro Recall@K*. Recall@K is a popular choice in most of the visual relation detection studies. The main reason is the incompleteness of visual relation detection datasets, i.e. some relations might not be annotated in the test set, while due to the model's generalization, they might get higher prediction values than the annotated ones. This sensitivity is handled by the K parameter in Recall@K.

b) Macro Recall@K: We define this metric as:

$$\text{MACRO RECALL@K} = \sum_{(s,p,o)\in\mathcal{T}_p} \frac{\text{MICRO R@K}(p)}{|\mathcal{T}_p|} \tag{3}$$

where $\mathcal{T}_p \subset \mathcal{T}$ is set of all relations with predicate p, and MICRO R@K(p) is computed on $\mathcal{T}_p$. The motivation behind this metric is the highly imbalanced distribution of classes in some datasets such as VG. In these datasets Micro Recall@K score gets dominated by frequently labeled relations and might not reflect the improvements in some important but under-represented classes. However, in Macro R@K, the prediction accuracy of under-represented classes can have a stronger effect on the output. This metric is inspired from the Macro F1 measure [29].

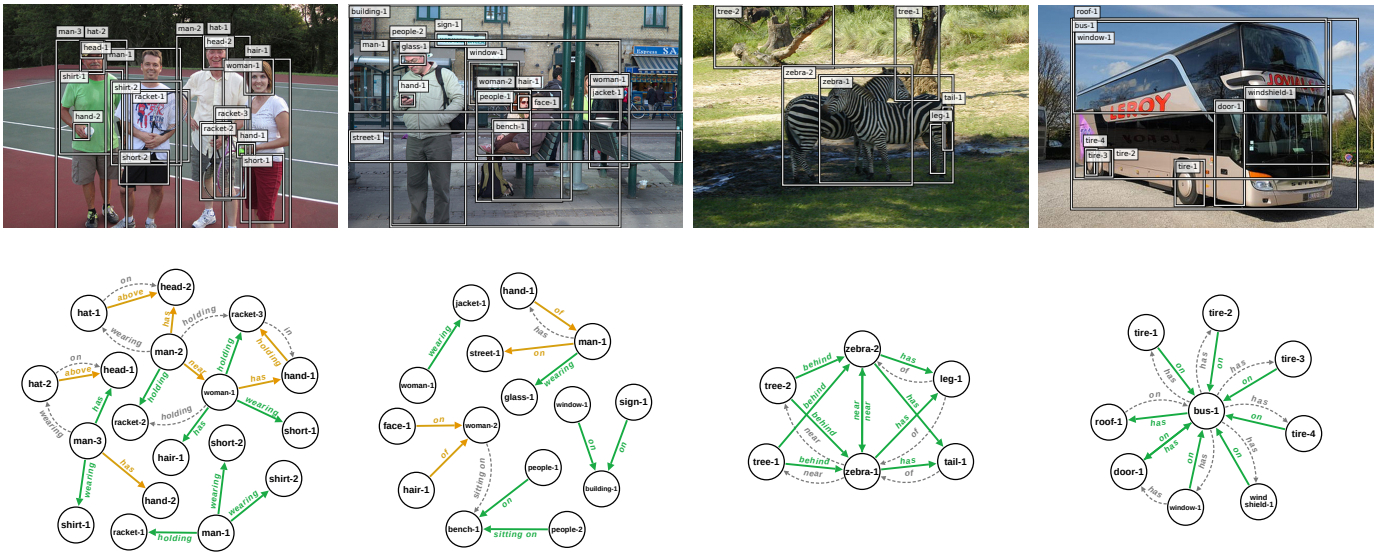

Fig. 3. Some of the qualitative results from our model's predictions. Green arrows indicate the successfully detected predicates (true positives), orange arrows indicate the false negatives and gray arrows indicate predicted links which are not annotated in the ground truth. (© 2020 IEEE)

C. Architectures

a) RGB-to-Depth Network: We employ the RGB-to-Depth architecture that has been introduced in [30]. The model is a fully convolutional neural network built on ResNet-50 [31], and trained in an end-to-end fashion on data from NYU Depth Dataset v2 [4]. In our experiments, we also trained the model from the outdoor images of Make3D dataset [32]. However, the model that was trained on this dataset, did not show promising results for relation detection. This observation is not surprising because unlike Visual Genome, Make3D images contain mostly outdoor scenes with very few objects.

b) RGB Feature Extraction: To extract embeddings and class probabilities of RGB images, we use the VGG-16 architecture [33] pre-trained on ImageNet [34] and fine-tuned on VG by Zellers et al. [18].

c) Depth Map Feature Extraction: For depth map extraction we use ResNet-18 proposed in [31]. We trained this model from scratch following the earlier discussions in Subsection III-A2. This network was trained separate from other inputs and on a pure depth-based, relation detection task using Adam [35], with a learning rate of 10^{-4} and batch size of 32 for 30 epochs.

d) Relation Detection Network: In relation detection head, each extracted feature pair goes to a separate, fully connected hidden layer of 64 neurons (∼12K learnable weights) for class probabilities, 512 for RGB feature maps (∼4M learnable weights), 4096 for depth feature maps (∼4M learnable weights) and 20 for location features (160 learnable weights). Each of them with a dropout rate of 0.1, 0.8, 0.6 and 0.1. The concatenated outputs are then connected to a fully connected hidden layer of 4096 neurons with 0.1 dropout and then to the classification layer. We trained this network by Adam [35], with a learning rate of 10^{-5}. We used a batch size of 16 and 30 epochs of training. All of the layers were initialized with Xavier weights [36].

D. Comparing Methods

We compare our results with *VTransE* [28] that takes visual embeddings and projects them to relation space using TransE. We also compare to the student network of [16] (*Yu's-S*), and their full model (*Yu's-S+T*) that employs external language data from Wikipedia. From the context propagating methods, we report Neural Motifs [18], Graph R-CNN [19] and IMP [17]. In an ablation study, we report our relation prediction results under several settings in which different combinations of object information are employed for prediction.

E. Experiments

As our main goal is to investigate the role of depth maps and other features in relation detection, we report *predicate prediction* results. In this setting, the relation detection performance is analyzed by isolating it from the object detector's error. Therefore, the goal is to evaluate the relation detection accuracy given the objects in an image. We carried on our experiments by training each model 8 times with different random seeds. The maximum variance of the results was no more than 0.01. The results are shown in Table I. In what follows, we provide a discussion over the quantitative and qualitative results.

The upper part of the table demonstrates the results directly reported from other works while the lower part presents the results from the ablation study on our model. For NM, we have computed the Macro R@K results using their publicly available code. We can see that our full model with depth maps, achieves the highest accuracy in comparison to the others in all settings. It is also interesting to note that when

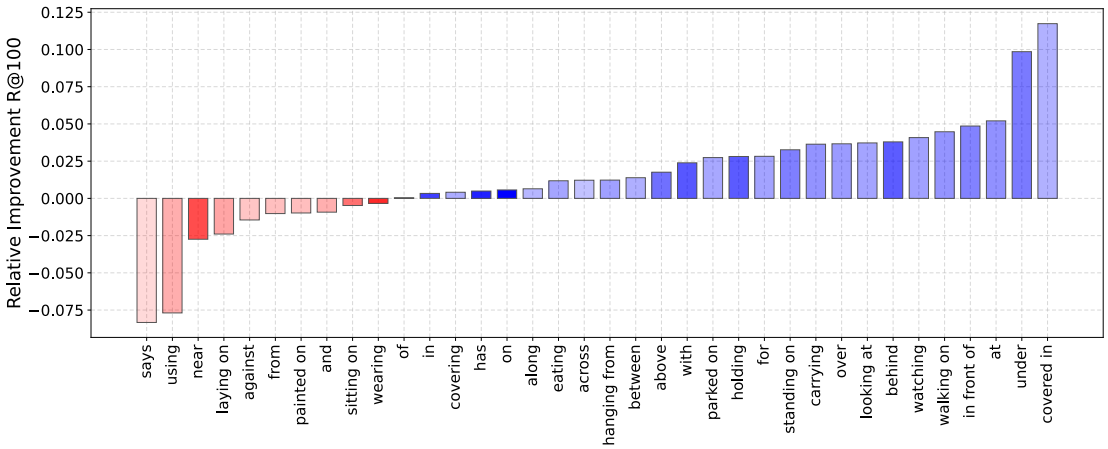

Fig. 4. This plot shows the prediction changes per predicate, going from *Ours-*v to *Ours-*v, d. The classes with zero changes are omitted from the plot. The darker shades indicate larger number of that class within the test set whereas the lighter shades are under-represented classes. An improvement in predicates with more frequency has a larger effect on the Micro R@K whereas this effect is eliminated within Macro R@K. We can see that indeed the improvements by using depth maps are mostly happening within the less-represented classes. (© 2020 IEEE)

using *only* depth maps we can already achieve a significant accuracy in predicate prediction, emphasizing the value of relational information that are stored within the depth maps alone. By comparing *Ours-*v to *Ours-*v, d, we can observe the improvements that depth maps bring. Also comparing *Ours-*l, c, d to *Ours-*l, c, v is specially informative from two aspects: (1) It shows that while some results are almost equal in Micro settings, one can observe a significant difference in the Macro setting, demonstrating the effectiveness of this metric in presenting the improvements of under-represented classes. (2) We observe that v alone has a higher R@K than d alone. However, when we add them separately to c, l we can see that d has more to offer. In other words, v brings more redundant information to c, l compared to d. To get a better intuition of the improvements that we gain after including depth maps (*Ours-*v, d compared to *Ours-*v), we plotted the changes in prediction accuracy for each predicate in Figure 4. We used darker shades for over-represented classes and lighter shades for under-represented ones. This helps to also gain a better intuition of improvement versus frequency of data. For example we can see that in general the accuracy of relations including the predicates such as `under`, `in front of` and `behind` has been improved. These predicates appear much less often in the dataset than `on` or `has`, having less effect in the computed *Micro* accuracy. Figure 5 presents some samples of synthetically generated depth maps in VG-Depth dataset including both high quality and faulty ones. Additionally, we present some of the predicted relations by our model in Figure 3.

V. Conclusion

We employed an RGB-to-Depth network, trained on a large corpus of data, to generate depth maps for Visual Genome dataset, releasing a new extension called *VG-Depth*. We provided a metric, *Macro R@K* for better evaluation of relation detection in Visual Genome and other highly imbalanced datasets. In extensive empirical evaluations, we demonstrated the effect of different object features in visual relation detection and showed that by using depth information, we achieve significantly better performance compared to other state-of-the-art methods.

VI. Acknowledgements

We thank Evgeniy Faerman, Vaheh Hatami, Alireza Ghazaei and the anonymous reviewers for their fruitful comments. This work was supported by the BMBF as part of the project MLWin (01IS18050).

References

[1] © 2020 IEEE. Reprinted, with permission, from S. Sharifadeh, S. Baharlou, M. Berrendorf, R. Koner,V. Tresp, "Improving visual relation detection using depth maps" in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 3597–3604.

[2] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European Conference on Computer Vision*. Springer, 2016, pp. 852–869.

[3] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[4] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[5] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data." in *ICML*, vol. 11, 2011, pp. 809–816.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2787–2795. [Online]. Available: http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf

[7] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[8] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International Conference on Machine Learning*, 2016, pp. 2071–2080.

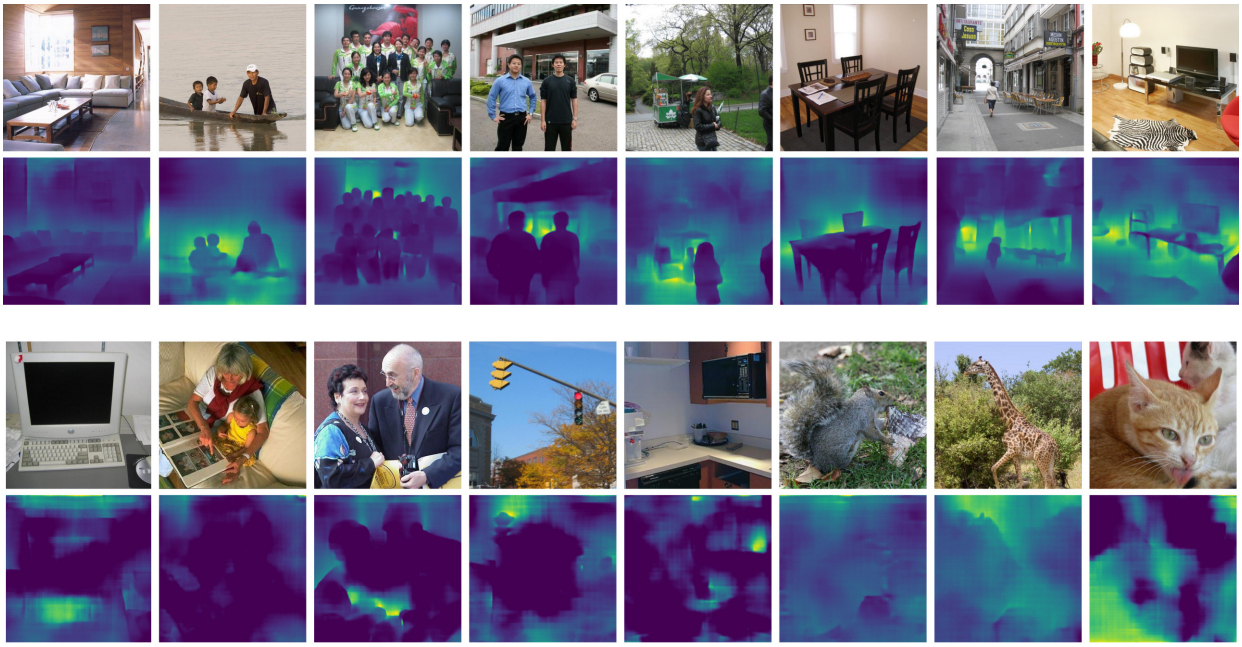

Fig. 5. The first two rows are the examples of visual genome images and their synthetically generated high quality depth maps. The second two rows are the examples of visual genome images and their synthetically generated noisy depth maps. (© 2020 IEEE)

[9] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 601–610.

[10] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2016.

[11] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in neural information processing systems*, 2013, pp. 926–934.

[12] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, and J. Lehmann, "Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework," *arXiv preprint arXiv:2006.13365*, 2020.

[13] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, and J. Lehmann, "Pykeen 1.0: A python library for training and evaluating knowledge graph emebddings," *arXiv preprint arXiv:2007.14175*, 2020.

[14] S. Baier, Y. Ma, and V. Tresp, "Improving visual relationship detection using semantic modeling of scene descriptions," in *International Semantic Web Conference*. Springer, 2017, pp. 53–68.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in neural information processing systems*, 2013, pp. 2787–2795.

[16] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, "Visual relationship detection with internal and external linguistic knowledge distillation," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5410–5419.

[18] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5831–5840.

[19] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 670–685.

[20] V. Tresp, S. Sharifzadeh, and D. Konopatzki, "A model for perception and memory."

[21] V. Tresp, S. Sharifzadeh, D. Konopatzki, and Y. Ma, "The tensor brain: Semantic decoding for perception and memory," *arXiv preprint arXiv:2001.11027*, 2020.

[22] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Experimental Robotics*. Springer, 2013, pp. 387–402.

[23] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 681–687.

[24] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 345–360.

[25] H.-K. Yang, A.-C. Cheng, K.-W. Ho, T.-J. Fu, and C.-Y. Lee, "Visual relationship prediction via label clustering and incorporation of depth information," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[27] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 213–228.

[28] H. Zhang, Z. Kyaw, S. Chang, and T. Chua, "Visual translation embedding network for visual relation detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 3107–3115. [Online]. Available: https://doi.org/10.1109/CVPR.2017.331

[29] H. Schütze, C. D. Manning, and P. Raghavan, "Introduction to information retrieval," in *Proceedings of the international communication of association for computing machinery conference*, 2008, p. 260.

[30] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 239–248.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[32] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from

a single still image," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

5 A Model for Perception and Memory

This chapter includes the following publication:

Tresp et al. [2019]

Declaration of Authorship The research idea was developed by Volker Tresp and further developed by Volker Tresp and Sahand Sharifzadeh and discussed with Dario Konopatzki. Volker Tresp and Sahand Sharifzadeh designed the experiments. Sahand Sharifzadeh implemented the code for all experiments. The manuscript was mainly written by Volker Tresp and Sahand Sharifzadeh wrote the section on experiments. The final manuscript was revised by all authors.

A Model for Perception and Memory

Volker Tresp, Sahand Sharifzadeh and Dario Konopatzki (firstname.lastname@lmu.de)
LMU Informatik, Oettingenstraße 67, 80538 München, Germany

Abstract

We analyze the close link between perception and memory. Our main hypothesis is that some of the main memory systems of the human brain, e.g., the episodic memory, the semantic memory, and to some degree also the working memory, are by-products of the need for humans to gradually extract more meaningful and more complex information from sensory inputs. Our model is an extension to the tensor memory approach. The key notions are index representations for entities, concepts, relationships and time instances, embeddings associated with the indices, a working memory layer, and a sensory memory layer. Perception and memory are realized as an interplay between the different layers. Our model is both competitive to other technical solutions and, as we argue, biologically plausible. Our experiments demonstrate that semantic memory can evolve from perception as a distinguishable functional module.

Introduction

Perception has evolved from simple stimulus-reaction in lower animals to the ability of a deep analysis of sensory input in humans. An important capability, for example, is the comparison to previous experiences: if a certain event is very similar to a past event, and that past event triggered a certain action, it makes sense that the current event should trigger the same action. Another important function is the identification of concepts and their relationships: "a child, located on a swing" will trigger very different actions than "a child, running in front of a car". Clearly a more refined perception is tightly linked to an improved understanding of the world, its schema, objects and their relationships, or as Goethe put it: "you only see what you know". In this paper we argue that episodic memory, i.e., the faculty to recall and restore past events, and semantic memory, i.e., knowledge about the world, are by-products of an evolving perceptual system which developed to deal with an increasingly complex world: our hypothesis is that episodic memory and semantic memory did not initially evolve as separate memory functions but instead repurposed faculties developed in perception for a semantic decoding of sensor stimuli. Furthermore, working memory might have evolved out of the need to store information to improve perceptual decoding.

The work in this paper is based on the tensor memory approach (Tresp et al., 2015; Tresp & Ma, 2016) which is an extension to the hippocampal memory indexing theory (Teyler & DiScenna, 1986). The key concepts of that approach are sparse index representations for entities, relationships and time instances. Each index has an associated distributed embedding, and memory and perception are based on an interplay between both. Perception, episodic memory and semantic memory might evoke sub-symbolic associations, but they are also declarative, indicated by the abilities of humans to report verbally about perception and memory contents. The semantic decoding in the tensor memory has exactly that declarative nature!

Here we significantly modify and extend that model. In the tensor memory model, the calculations of conditional probabilities required for decoding require marginalization operations which are costly and might be difficult to realize with biological wetware. Also, several indices and their embeddings needed to be active at the same time, which might not be biologically plausible (binding problem) and the approach required units to implement multiplication. Here, we propose a layered approach, where the sensory information is processed by a working memory layer, a representation layer and an index layer. The operations can be described as a single recurrent neural network where semantic memory evolves as an identifiable functional module.

The remaining parts of the paper are organized as follows. After we provide a brief review of the tensor memory approach in the next section, we present our model and mathematical operations performed by the model. Then follows a discussion on the neural substrate and a presentation of experimental results. The last section contains our conclusions.

Tensor Memories

Triple-based graphs have evolved into major data structures for representing semantic information. Concrete examples are knowledge graphs which store world facts (e.g., *(Munich, partOf, Bavaria)*) and scene graphs for describing image content (e.g., in the actual image, *(Dog, bites, Person)*).[1] The graphs are based on *(s, p, o)*-triples where the subject *s* and the object *o* are entities represented as the nodes in the graph, and where a directed link, labeled by *p*, represents a predicate. In the tensor memory approach, a graph was represented as a 3-way tensor, which was approximated by tensor factorization involving latent embeddings as vectors of real numbers: $\mathbf{a}_{e_s}$ is the embedding associated with the subject, $\mathbf{a}_{e_o}$ is the embedding associated with the object, $\mathbf{a}_p$ is the embedding associated with the predicate, and $\mathbf{a}_t$ is the embedding associated with the time instance, or image, t. Note that an entity has a unique representation, independent of its role as a subject or object. The factorized models deliver estimates for the probability that a triple is true at time t, given image information at time t, i.e., $P(s,p,o|t)$, and $P(s,p,o)$, which is the

[1] The nodes in the graph represent entities. In a knowledge graph, the nodes are labeled by identifiers (*Jack*), in scene graphs by concept labels (*Person*).

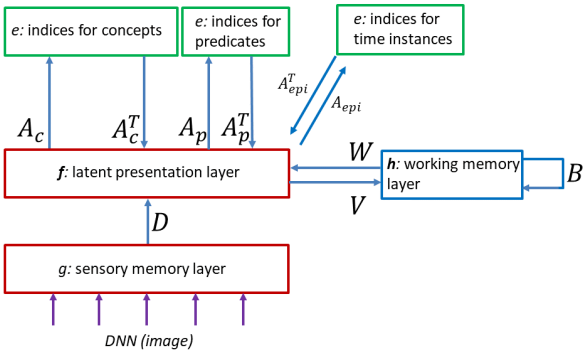

Figure 1: Our model architecture consists of four layers. Extracted representations from images are represented at the bottom layer (sensory memory, $\mathbf{g}$) which is connected to the representation layer $\mathbf{f}$. The top layer $\mathbf{e}$ contains the indices for concepts, predicates, and time instances. The working memory $\mathbf{h}$ is an integration layer and $\mathbf{g}$ is the sensory layer.

prior probability for observing the triples (s, p, o).[2]

The tensor memory model has some technical shortcomings when used in perception. For example, the semantic memory was derived from a marginalization over time, which is a computationally expensive operation that might not easily be implemented in biological wetware and can only be executed efficiently for some models (Tresp & Ma, 2016). Other problems are the polynomial scaling with the rank of the tensor model and the need for units that can perform multiplications.

A Model for Perception and Memory

A Layered Architecture: Figure 1 shows our model architecture. As in the tensor memory model, we assume an **index representation layer** $\mathbf{e}$ for entities, predicates and time instances, which is shown at the top of the figure. The indices can activate the **representation layer** $\mathbf{f}$ via connection matrix A_c^T for the concepts, A_p^T for the predicates, and A_{epi}^T for time instances. The embedding of concept e_i is the vector $\mathbf{a}_{e_i}$, which is the transpose of the i-th row of A_c. Similar for the predicates and the time instances. When index e_i is active and all other indices are inactive, then $\mathbf{f} = \mathbf{a}_i$. We introduce the **working memory layer** $\mathbf{h}$. This layer has some internal dynamics and receives inputs from the representation layer $\mathbf{f}$. In the following, we assume that we want to retrieve two concepts and their relationships at time, or image, t. Let t be the time constant of perception (on the order of hundreds of milliseconds). The micro time-step τ is the time constant for the decoding of the sensory input ($\tau << 100ms$). We now discuss the individual processing steps.

[2]More explicitly, $P(s, p, o|t)$ stands for the probability of observing a subject entity and an object entity at time t, where the subject belongs to concept s, the object belongs to concept o, and both are related by predicate p.

Decoding the Subject: Consider that $\mathbf{g}(t)$ is the embedding of the sensory input at time t. The activations of the working memory become, with $\mathbf{h}_{in}(t) = 0$,

$$\mathbf{h}(t) = sig(\mathbf{h}_{in}(t) + VD\mathbf{g}(t)).$$

The activations in the representation layer and the index layer are calculated as

$$\mathbf{f}(t) = D\mathbf{g}(t) + W\mathbf{h}(t) \quad and \quad \mathbf{e}(t) = sig(A_c\mathbf{f}(t)).$$

Thus the activations of the indices are determined by the inner product of their embeddings with the activation of the representation layer. In training, $\mathbf{e}(t)$ is set to be a one-hot vector indicating the index of the true subject. In testing, we proceed with $\mathbf{e}(t)$.[3] Finally, we set,

$$\mathbf{f}(t) \leftarrow A_c^T\mathbf{e}(t) = \mathbf{a}_{e_s} \quad and \quad \mathbf{h}_{in}(t+\tau) = B\mathbf{h}(t) + V\mathbf{a}_{e_s}.$$

In training, $\mathbf{f}(t)$ is now set to be the embedding of the true subject e_s, and in testing, it is an average, weighted by $\mathbf{e}(t)$; $\mathbf{h}_{in}(t+\tau)$ is the input activation for the working memory in the next time step. All weight matrices D, V, W, B and the matrices containing the embeddings A_c, A_p, A_{epi} are learned in training. Note that here, and in the following, there is a direct short cut, not involving the potentially slower working memory, in the form of $\mathbf{e}(t) = sig(A_cD\mathbf{g}(t))$.

Decoding the Object: The object decoding is identical to the subject decoding, if we replace t with $t+\tau$, $t+\tau$ by $t+2\tau$, and $\mathbf{a}_{e_s}$ by $\mathbf{a}_{e_o}$.

Decoding the Predicate: The predicate decoding is identical to the subject decoding, if we replace t with $t+2\tau$, $t+\tau$ by $t+3\tau$, $\mathbf{a}_{e_s}$ by $\mathbf{a}_p$, and A_c by A_p. Note that the decoding is asymmetrical and can distinguish between *(Dog, bites, Person)* and *(Person, bites, Dog)*. For a given image, the decoding can generate a large number of triples, which, in their entirety, present a visual input as an ensemble scene graph.

Discussion

Sensory Memory Layer: $\mathbf{g}$ is the visual sensory memory, maintaining visual information to be processed and analyzed. $\mathbf{g}$ represents properties of the respective focus of attention (in technical systems, these would be the bounding boxes). We assume that sensor processing involves an attention mechanism, such that $\mathbf{g}(t)$ represents the subject bounding box, $\mathbf{g}(t+\tau)$ represents the object bounding box, and $\mathbf{g}(t+2\tau)$ represents the predicate bounding box. The latter includes the two previous bounding boxes and some surrounding image area. In the brain, it is assumed that the sensory memory layer involves the visuospatial sketchpad of the working memory, associated with the parietal-occipital region.

[3]In testing we could perform a sampling from a normalized version of $\mathbf{e}(t)$; but this sampling introduces noise and would have to be repeated many times; proceeding with $\mathbf{e}(t)$ can be considered an approximation to the sampling.

Index Layer: The index layer $\mathbf{e}$ consists of indices for concepts, like *Cat*, and predicates like *nextTo*, and time instances. Generally it is assumed that indices are formed in the hippocampus and their long-term representation might involve the pole of the temporal lobe. An index might be realized by a small number of interacting neurons (Teyler & DiScenna, 1986; Quiroga, 2012). Over the path $\mathbf{e} \rightarrow \mathbf{f} \rightarrow \mathbf{g}$, an index can also excite a sensory impression. The indices (including the indices for time instances) have a relational memory function in the sense that they bind together different dimensions in the representation layer.

Representation Layer: The representation layer is important for the information path from $\mathbf{g}$ to $\mathbf{e}$ and it interacts with the working memory $\mathbf{h}$. If index e_i is activated, the activation of layer $\mathbf{f}$ reflects $\mathbf{a}_i$. Thus, whereas the sensory layer is primarily visually grounded, the representation layer is primarily concept grounded. If the concept "cat" is active in the index layer, the representation layer would contain abstract representations of the concept cat, without a reference to the actual cat in the sensory input. In the brain, these representations might involve the parietal lobe and the posterior region of the temporal lobe.

Working Memory Layer: The working memory layer integrates information from visual input and the decoding process (*subject, predicate, object*), and eventually the complete scene with its visual representations and decoded concepts and predicates. Working memory might have initially been developed biologically to support a more complex scene understanding and event processing. Its integrative functions are typically associated with the prefrontal cortex (PFC) in the frontal lobe and its interaction with the representation layer might reflect the event-specific relational memory functions in perception and memory recall. The PFC is profusely and reciprocally connected with the hippocampus, and cortices of association of the temporal and parietal lobes. Note that this layer is the "intelligence on top", since a simpler decoding $\mathbf{g} \rightarrow \mathbf{f} \rightarrow \mathbf{e}$ would not involve the working memory layer.

Semantic Decoding, Schema, and Semantic Memory: Whereas the restoration of an episodic memory trace is mostly sub-symbolic and might lead to an autonoetic experience, our model also contains a semantic decoding for perception and episodic memory. It produces a set of triples on a symbolic level involving indices for concepts and predicates and their embeddings, which are encoded as connection patterns (Tresp et al., 2015). In the cognitive sciences, representations for concepts form what is called a *schema*, which aids in the interpretation of events. Studies have shown that individuals can analyze perceptual information significantly more easily when this information is related to an acquired schema. According to our model, an improvement in the schema would go hand in hand with a refined perception. (Moscovitch et al., 2016) defines a schema as "adaptable associative networks of knowledge extracted over multiple similar experiences", which is in agreement with our model. The same paper states that "memories for recent events draw on interactions between schemas, semantics, and perceptual aspects of an experience, mediated in part by different regions in the anterior and posterior neocortex", which we would interpret as the multilevel processing in our model.

Early in evolution, it was important for individuals to recognize particular classes of objects (e.g., "tigers", "snakes"); object recognition then became the basis for a more meaningful information extraction in form of semantic triples. Our model requires a storage layer which maintains information about already extracted concepts; as proposed already, this storage might have been the initial motivation for the brain to evolutionary develop a working memory in the PFC.

Another by-product in our approach is semantic memory. In our model, semantic memory uses the same layered structure, ignoring the sensory input, and models the prior probability for observing a triple. Thus semantic memory involves only the top three layers and is independent of the context provided by the sensory input. Assume the index for *Cat* is activated in the index layer by some internal or external cue. Then, without any perceptual input, the decoding process might generate, with some probability, that *(Cat, sitsOn, Stove)*. Mathematically, the semantic memory here models $P(p = \textit{sitsOn}, o = \textit{Stove} | s = \textit{Cat})$. In our model, the semantic memory is implemented as the connection pattern between the index layer and the representation layer. In the brain, semantic memory involves the anterior temporal cortex (Moscovitch et al., 2016).

A scene graph describes entities and their relationships. So far we focused on the concept attributes of the entities: *(Dog, bites, Person)* and not identifier attributes as in *(Sparky, bites, Jack)*. Humans have an enormous capacity to represent a large number of entities; but consider a less complex mammal which needs to have only knowledge about a smaller number of specific entities, such as the leader hierarchy in a pack. We propose that, for significant entities, indices are formed as well. So in the previous example, there would be indices for *Jack* and *Sparky*, in addition to the indices for *Person* and *Dog*.

Our model does not explicitly consider properties like *large, red, threatening*. These can be treated as concepts in conjunction with the predicate *hasAttribute* where the visual information for subject and object originate from an identical image region. Also the representations in the sensory layer and in the representation layer might convey attribute information.

Episodic Memory: Most researchers consider temporal coding to be a core function of the hippocampus and not a derived property (Teyler & DiScenna, 1986; Eichenbaum, 2014; Moscovitch et al., 2016). Our model agrees with this view and we assume that an index for a time instance is formed for a sensory input that is associated with an emotion or with novelty (Figure 1). In its simplest form, the t-th row of the matrix A_{epi} copies $\mathbf{f}$. Biologically, time indices might involve a small network of interacting neurons (Quiroga, 2012); together with their connection patterns (in our model A_{epi}) they form mem-

ory traces or engrams. It is assumed that the original purpose of this index was to be able to compare the current event to previously encountered events (familiarity) and their associated actions, supporting the individual in decision making. In the course of evolution, this decision oriented process was repurposed and various cues were able to activate the indices which, using bidirectional connections, are then able to restore a past memory as a personal experience. Subsequently, this function became more elaborate and enabled future-oriented mental time travel to evaluate future consequences of actions. Humans became able to mentally place themselves in the past, in the future, or in counterfactual situations, a process called autonoetic consciousness. Episodic memory traces can also be used to train implicit memories in perceptual and procedural memories or even train complex action patterns (Kumaran, Hassabis, & McClelland, 2016). An episodic memory experience is an active process that involves details of the event and its location (Moscovitch et al., 2016). Sometimes the reconstruction is considered a Bayesian process of reconstructing the past as accurately as possible based on the engram information. According to the standard consolidation theory, indices are consolidated in neocortex, whereas the multiple trace theory proposes that the hippocampal representation maintains its function over long periods and a memory trace is only partially consolidated in neocortex (Moscovitch et al., 2016). In our model, consolidation would involve a reimplementation of an index and its connection pattern.

Experiments

We use the Stanford Visual Relationship data set, which is the basis for many works on scene analysis, e.g., (Baier, Ma, & Tresp, 2017). We used 100 concepts and 70 predicates with 4000 images for training and 1000 for testing. The results of our model are comparable to highly optimized models in other works (Table 1). We also see that the working memory is essential for obtaining good results. The dimensions for the layers are $\mathbf{g}/4096, \mathbf{f}/4096, \mathbf{h}/500$. For comparison, we report results from (Baier et al., 2017).

We also did experiments where we removed the visual inputs and our model performed as a semantic memory. The table shows that the performance of this derived model is worse than a model optimized on semantic data (Baier) but much better than random. The table also shows that by starting with a perception model (trained on 10 epochs) and then adding (1 or 9) epochs, where we only use the semantic triples without perceptual input, significantly improves the extracted semantic model with only a small performance drop in perception.

Conclusion

We have presented a mathematical model for perception, episodic memory and semantic memory, and related it to cognitive models of the human brain. Our main hypothesis is that episodic memory, semantic memory, and to some degree also working memory, are by-products of the need for humans to extract more meaningful and more complex information from sensory inputs. We could show experimentally that semantic memory can evolve as a by-product of perception. The semantic memory represents prior probabilities, which might be an interesting basis for a Bayesian brain interpretation. We propose that the model we presented is in a sense minimalist, containing necessary perceptual components.

Table 1: ph stands for phrase detection and pr stands for predicate detection. In phrase detection, a triple with its corresponding bounding boxes is considered a success, if both the triple and the bounding boxes are correctly detected. In predicate detection, subject concept and object concept are given and the task is to predict the predicate (Baier et al., 2017). For z-s-ph/z-s-ph (zero shot), we only evaluate the test set performance on triples that did not occur in training. The first row (Model) shows results for our model. The fourth row (Baier) shows the results from literature. Dir are results where we removed the working memory. Our model gives better results for the zero-shot experiments. The last two columns report recall results for only the semantic memory. The first row shows results where the semantic memory was extracted from our perceptual model. The result (82.46 and 53.53) are worse than the result for Baier, where the latter was trained directly on the triple data. S1 and S9 show results where we added 1 and 9 epochs of pure semantic training to the perception model. We see that the semantic model improves significantly with almost no degradation on perception.

Method	ph	z-s-ph	pr	z-s-pr	@10	@1
Model	23.45	**10.95**	93.32	78.79	82.46	53.53
S1	23.32	10.44	93.17	**80.07**	93.46	67.55
S9	22.61	9.24	92.77	79.47	94.77	68.68
Baier	**25.11**	7.96	**93.81**	76.05	**95.86**	**70.50**
Dir	11.13	7.87	77.19	65.61	-	-
Rand	0.01	0.00	18.53	16.51	0.08	0.01

References

Baier, S., Ma, Y., & Tresp, V. (2017). Improving visual relationship detection using semantic modeling. In *ISWC*.

Eichenbaum, H. (2014). Time cells in the hippocampus. *Nature Reviews Neuroscience*, *15*(11).

Kumaran, D., Hassabis, D., & McClelland, J. L. (2016). What learning systems do intelligent agents need? *Trends in Cognitive Sciences*, *20*(7), 512–534.

Moscovitch, M., et al. (2016). Episodic memory and beyond. *Annual review of psychology*.

Quiroga, R. Q. (2012). Concept cells: the building blocks of declarative memory functions. *Nat Rev Neurosci*, *13*(8).

Teyler, T. J., & DiScenna, P. (1986). The hippocampal memory indexing theory. *Behavioral neuroscience*.

Tresp, V., & Ma, Y. (2016). The tensor memory hypothesis. In *Nips workshop on representation learning)*.

Tresp, V., et al. (2015). Learning with memory embeddings. *NIPS Workshop on Representation Learning*.

6 An unsupervised joint system for text generation from knowledge graphs and semantic parsing

This chapter includes the following publication:

Schmitt et al. [2019]

and the code is made public as

`https://github.com/mnschmit/unsupervised-graph-text-conversion`

Declaration of Authorship The initial idea was brainstormed by Sahand Sharifzadeh and Martin Schmitt. The idea was further developed by Martin Schmitt. Martin Schmitt performed the implementations and evaluations. The final manuscript was mainly written by Martin Schmitt and revised by all authors.

An Unsupervised Joint System for Text Generation from Knowledge Graphs and Semantic Parsing

Martin Schmitt[1] **Sahand Sharifzadeh**[2] **Volker Tresp**[2,3] **Hinrich Schütze**[1]

[1]Center for Information and Language Processing (CIS), LMU Munich
[2]Department of Informatics, LMU Munich
[3]Siemens AG Munich
martin@cis.lmu.de

Abstract

Knowledge graphs (KGs) can vary greatly from one domain to another. Therefore supervised approaches to both graph-to-text generation and text-to-graph knowledge extraction (semantic parsing) will always suffer from a shortage of domain-specific parallel graph-text data; at the same time, adapting a model trained on a different domain is often impossible due to little or no overlap in entities and relations. This situation calls for an approach that (1) does not need large amounts of annotated data and thus (2) does not need to rely on domain adaptation techniques to work well in different domains. To this end, we present the *first approach to unsupervised text generation from KGs* and show simultaneously how it can be used for *unsupervised semantic parsing*. We evaluate our approach on WebNLG v2.1 and a new benchmark leveraging scene graphs from Visual Genome. Our system outperforms strong baselines for both text↔graph conversion tasks without any manual adaptation from one dataset to the other. In additional experiments, we investigate the impact of using different unsupervised objectives.[1]

1 Introduction

Knowledge graphs (KGs) are a general-purpose approach for storing information in a structured, machine-accessible way (Van Harmelen et al., 2008). They are used in various fields and domains to model knowledge about topics as different as lexical semantics (Fellbaum, 2005; van Assem et al., 2006), common sense (Speer et al., 2017; Sap et al., 2019), biomedical research (Wishart et al., 2018) and visual relations in images (Lu et al., 2016).

This ubiquity of KGs necessitates interpretability because diverse users – both experts and non-experts – work with them. Even though, in principle, a KG is human-interpretable, non-experts may have difficulty making sense of it. Thus, there is a need for methods, such as automatic natural language generation ("graph→text"), that support them.

Semantic parsing, i.e., the conversion of a text to a formal meaning representation, such as a KG, ("text→graph") is equally important because it makes information that only exists in text form accessible to machines, thus assisting knowledge base engineers in KG creation and completion.

As KGs are so flexible in expressing various kinds of knowledge, separately created KGs vary a lot. This unavoidably leads to a shortage of training data for both graph↔text tasks. We therefore propose an unsupervised model that (1) easily adapts to new KG domains and (2) only requires unlabeled (i.e., non-parallel) texts and graphs from the target domain, together with a few fact extraction heuristics, but no manual annotation.

To show the effectiveness of our approach, we conduct experiments on the latest release (v2.1) of the WebNLG corpus (Shimorina and Gardent, 2018) and on a new benchmark we derive from *Visual Genome* (Krishna et al., 2016). While both of these datasets contain enough annotations to train supervised models, we evaluate our unsupervised approach by ignoring these annotations. The datasets are particularly well-suited for our evaluation as both graphs and texts are completely human-generated. Thus for both our tasks, models are evaluated with natural, i.e., human-generated targets.

Concretely, we make the following contributions: (1) We present the first unsupervised non-template approach to text generation from KGs (graph→text). (2) We jointly develop a new unsupervised approach to semantic parsing that automatically adjusts to a target KG schema (text→graph). (3) In contrast to prior unsupervised graph→text and text→graph work, our model does not re-

[1]https://github.com/mnschmit/unsupervised-graph-text-conversion

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7117–7130,
November 16–20, 2020.

quire manual adaptation to new domains or graph schemas. (4) We provide a thorough analysis of the impact of different unsupervised objectives, especially the ones we newly introduce for text↔graph conversion. (5) We create a new large-scale dataset for text↔graph transformation tasks in the visual domain.

2 Related Work

graph → text. Our work is the first attempt at fully unsupervised text generation from KGs. In this respect it is only comparable to traditional rule- or template-based approaches (Kukich, 1983; McRoy et al., 2000). However, in contrast to these approaches, which need to be manually adapted to new domains and KG schemas, our method is generally applicable to all kinds of data without modification.

There is a large body of literature about supervised text generation from structured data, notably about the creation of sports game summaries from statistical records (Robin, 1995; Tanaka-Ishii et al., 1998). Recent efforts make use of neural encoder-decoder mechanisms (Wiseman et al., 2017; Puduppully et al., 2019). Although text creation from relational databases is related and our unsupervised method is, in principle, also applicable to this domain, in our work we specifically address text creation from graph-like structures such as KGs.

One recent work on supervised text creation from KGs is (Bhowmik and de Melo, 2018). They generate a short description of an entity, i.e., a single KG node, based on a set of facts about the entity. We, however, generate a description of the whole KG, which involves multiple entities and their relations. Koncel-Kedziorski et al. (2019) generate texts from whole KGs. They, however, do not evaluate on human-generated KGs but automatically generated ones from the scientific information extraction tool SciIE (Luan et al., 2018). Their supervised model is based on message passing through the topology of the incidence graph of the KG input. Such graph neural networks (Kipf and Welling, 2017; Veličković et al., 2018) have been widely adopted in supervised graph-to-text tasks (Beck et al., 2018; Damonte and Cohen, 2019; Ribeiro et al., 2019, 2020).

Even though Marcheggiani and Perez-Beltrachini (2018) report that graph neural networks can make better use of graph input than RNNs for supervised learning, for our unsupervised approach we follow the line of research that uses RNN-based sequence-to-sequence models (Cho et al., 2014; Sutskever et al., 2014) operating on serialized triple sets (Gardent et al., 2017b; Trisedya et al., 2018; Gehrmann et al., 2018; Castro Ferreira et al., 2019; Fan et al., 2019). We make this choice because learning a common semantic space for both texts and graphs by means of a shared encoder and decoder is a central component of our model. It is a nontrivial, separate research question whether and how encoder-decoder parameters can effectively be shared for models working on both sequential and non-sequential data. We thus leave the adaptation of our approach to graph neural networks for future work.

text → graph. Converting a text into a KG representation, our method is an alternative to prior work on open information extraction (Niklaus et al., 2018) with the advantage that the extractions, though trained without labeled data, automatically adjust to the KGs used for training. It is therefore also related to relation extraction in the unsupervised (Yao et al., 2011; Marcheggiani and Titov, 2016; Simon et al., 2019) and distantly supervised setting (Riedel et al., 2010; Parikh et al., 2015). However, these systems merely predict a single relation between two given entities in a single sentence, while we translate a whole text into a KG with potentially multiple facts.

Our text→graph task is therefore most closely related to semantic parsing (Kamath and Das, 2019), but we convert statements into KG facts whereas semantic parsing typically converts a question into a KG or database query. Poon and Domingos (2009) proposed the first unsupervised approach. They, however, still need an additional KG alignment step, i.e., are not able to directly adjust to the target KG. Other approaches overcome this limitation but only in exchange for the inflexibility of manually created domain-specific lexicons (Popescu et al., 2004; Goldwasser et al., 2011). Poon (2013)'s approach is more flexible but still relies on preprocessing by a dependency parser, which generally means that language-specific annotations to train such a parser are needed. Our approach is end-to-end, i.e., does not need any language-specific preprocessing during inference and only depends on a POS tagger used in the rule-based text→graph system to bootstrap training.

Unsupervised sequence generation. Our unsu-

pervised training regime for both text↔graph tasks is inspired by (Lample et al., 2018b). They used self-supervised pretraining and backtranslation for unsupervised translation from one language to another. We adapt these principles and their noise model to our tasks, and introduce two new noise functions specific to text↔graph conversion.

3 Preliminaries

3.1 Data structure

We formalize a KG as a labeled directed multigraph (V, E, s, t, l) where entities are nodes V and edges E represent relations between entities. The lookup functions $s, t : E \to V$ assign to each edge its source and target node. The labeling function l assigns labels to nodes and edges where node labels are entity names and edge labels come from a predefined set $\mathcal{R}$ of relation types.

An equivalent representation of a KG is the set of its facts. A fact is a triple consisting of an edge's source node (the subject), the edge itself (the predicate), and its target node (the object). So the set of facts $\mathcal{F}$ of a KG can be obtained from its edges:

$$\mathcal{F} := \{\, (s(e), e, t(e)) \mid e \in E \,\}.$$

Applying l to all triple elements and writing out $\mathcal{F}$ in an arbitrary order generates a serialization that makes the KG accessible to sequence models otherwise used only for text. This has the advantage that we can train a sequence encoder to embed text and KGs in the same semantic space. Specifically, we serialize a KG by writing out its facts separated with end-of-fact symbols (`EOF`) and elements of each fact with special `SEP` symbols. We thus define our task as a sequence-to-sequence (seq2seq) task.

3.2 Scene Graphs

The Visual Genome (VG) repository is a large collection of images with associated manually annotated **scene graphs**; see Fig. 1. A scene graph formally describes image objects with their attributes, e.g., (hydrant, `attr`, yellow), and their relations to other image objects, e.g., (woman, in, shorts). Each scene graph is organized into smaller subgraphs, known as **region graphs**, representing a subpart of a more complex larger picture that is interesting on its own. Each region graph is associated with an English text, the **region description**. Texts and graphs were not automatically produced from each other, but were collected from crowdworkers who

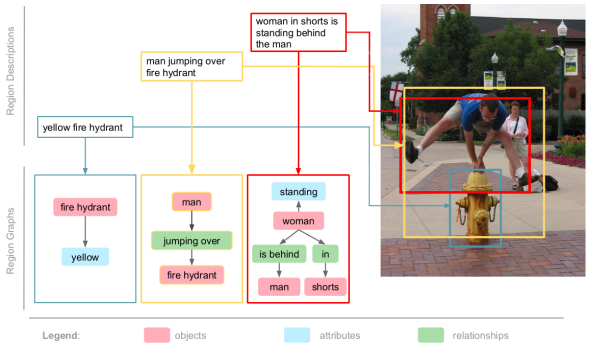

Figure 1: Region graphs and textual region descriptions in Visual Genome (VG). Image regions serve as common reference for text and graph creation but are disregarded in our work. We solely focus on the pairs of corresponding texts and graphs. Illustration adapted from (Krishna et al., 2016).

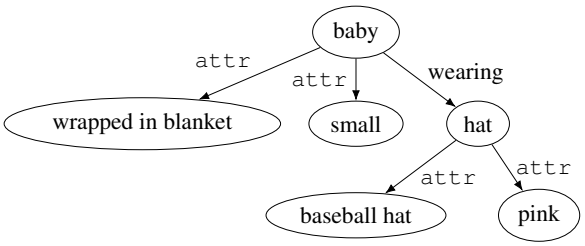

Figure 2: Example graph in our new VG benchmark.

were presented an image region and then generated text and graph. So although the graphs were not specifically designed to closely resemble the texts, they describe the same image region. This semantic correspondence makes scene graph↔text conversion an interesting and challenging problem because text and graph are not simple translations of each other.

Scene graphs are formalized in the same way as other KGs: V here contains image objects and their attributes, and $\mathcal{R}$ contains all types of visual relationships and the special label `attr` for edges between attribute and non-attribute nodes. Fig. 2 shows an example.

VG scene graphs have been used before for traditional KG tasks, such as KG completion (Wan et al., 2018), but we are the first to use them for a text↔graph conversion dataset.

4 Approaches

4.1 Rule-based systems

We propose a rule-based system as unsupervised baseline for each of the text↔graph tasks. Note that they both assume that the texts are in English.

$\mathbf{R}^{\text{graph}\to\text{text}}$. From a KG serialization, we remove

noise function	behavior
`swap`	applies a random permutation σ of words or facts with $\forall i \in \{1, \ldots, n\}, \lvert\sigma(i) - i\rvert \leq k$; $k = 3$ for text, $k = +\infty$ for knowledge graphs.
`drop`	removes each fact/word with a probability of p_{drop}.
`blank`	replaces each fact/word with a probability of p_{blank} by a special symbol `blanked`.
`repeat`	inserts repetitions with a probability of p_{repeat} in a sequence of facts/words.
`rule`	generates a noisy translation by applying $\text{R}^{\text{graph}\to\text{text}}$ to a graph or $\text{R}^{\text{text}\to\text{graph}}$ to a text.

Table 1: Noise functions and their behavior on graphs and texts.

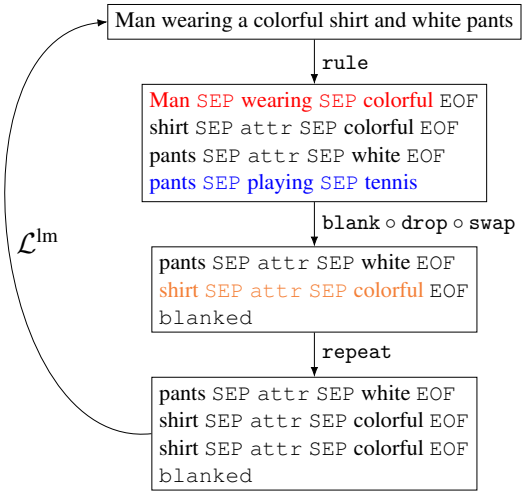

Figure 3: Example noisy training instance for the graph→text task in the composed noise setting. The fact highlighted in red is removed by `drop`, the one in blue is replaced with `blanked` by `blank`, the one in orange is repeated by `repeat`.

`SEP` symbols and replace `EOF` symbols by the word *and*. The special label `attr` is mapped to *is*. This corresponds to a template-based enumeration of all KG facts. See Table 5 for an example.

$\text{R}^{\text{text}\to\text{graph}}$. After preprocessing a text with NLTK's default POS tagger (Loper and Bird, 2004) and removing stop words, we apply two simple heuristics to extract facts: (1) Each verb becomes a predicate; *is* creates facts with predicate `attr`. The content words directly before and after such a predicate word become subject and object. (2) Adjectives a form attributes, i.e., build facts of the form $(X, \texttt{attr}, a)$ where X is filled with the first noun after a. These heuristics are similar in nature to a rudimentary parser. See Table 8 for an example.

4.2 Neural seq2seq systems

Our main system is a neural seq2seq architecture. We equip the standard encoder-decoder model with attention (Bahdanau et al., 2014) and copy mechanism (Gu et al., 2016). Allowing the model to directly copy from the source to the target side is beneficial in data to text generation (Wiseman et al., 2017; Puduppully et al., 2019). The encoder (resp. decoder) is a bidirectional (resp. unidirectional) LSTM (Hochreiter and Schmidhuber, 1997). Dropout (Hinton et al., 2012) is applied at the input of both encoder and decoder (Britz et al., 2017). We combine this model with the following concepts:

Multi-task model. In unsupervised machine translation, systems are trained for both translation directions (Lample et al., 2018b). In the same way, we train our system for both conversion tasks text↔graph, sharing encoder and decoder. To tell the decoder which type of output should be produced (text or graph), we initialize the cell state of the decoder with an embedding of the desired output type. The hidden state of the decoder is initialized with the last state of the encoder as usual.

Noisy source samples. Lample et al. (2018a) introduced denoising auto-encoding as pretraining and auxiliary task to train the decoder to produce well-formed output and make the encoder robust to noisy input. The training examples for this task consist of a noisy version of a sentence as source and the original sentence as target. We adapt this idea and propose the following noise functions for the domains of graphs and texts: `swap`, `drop`, `blank`, `repeat`, `rule`. Table 1 describes their behavior. `swap`, `drop` and `blank` are adapted from (Lample et al., 2018a) with facts in graphs taking the role of words in text. As order should be irrelevant in a set of facts, we drop the locality constraint in the `swap` permutation for graphs by setting $k = +\infty$.

Denoising samples generated by `repeat` requires the model to learn to remove redundant information in a set of facts. In the case of text, `repeat` mimics a behavior often observed with insufficiently trained neural models, i.e., repeating words considered important.

Unlike the other noise functions, `rule` does not "perturb" its input, but rather noisily backtranslates

it. We will see in Section 7 that bootstrapping with these noisy translations is essential.

We consider two fundamentally different noise injection regimes: (1) The **composed noise** setting is an adaptation of Lample et al. (2018a)'s noise model (`blank`∘`drop`∘`swap`) where our newly introduced noise functions `rule` and `repeat` are added to the start and end of the pipeline, i.e., all data samples are treated equally with the same noise function $C_{\text{comp}} :=$ `repeat`∘`blank`∘`drop`∘`swap`∘`rule`. Figure 3 shows an example. (2) In the **sampled noise** setting, we do not use all noise functions at once but sample a single one per data instance.

4.3 Training regimes

We denote the sets of graphs and corresponding texts by $\mathcal{G}$ and $\mathcal{T}$. The set of available supervised examples $(x, y) \in \mathcal{G} \times \mathcal{T}$ is called $\mathcal{S} \subset \mathcal{G} \times \mathcal{T}$. P_g and P_t are probabilistic models that generate, conditioned on any input, a graph (g) or a text (t).

Unsupervised training. We first obtain a language model for both graphs and text by training one epoch with the denoising auto-encoder objective:

$$\mathcal{L}^{\text{denoise}} = \mathop{\mathbb{E}}_{x\sim\mathcal{G}}[-\log P_g(x|C(x))] + \mathop{\mathbb{E}}_{y\sim\mathcal{T}}[-\log P_t(y|C(y))]$$

where $C \in \{C_{\text{comp}}\}$ for composed noise and $C \in \{\texttt{swap}, \texttt{blank}, \texttt{drop}, \texttt{repeat}, \texttt{rule}\}$ for sampled noise. In this pretraining epoch only, we use all possible noise functions individually on all available data. As sampled noise incorporates five different noise functions and composed noise only one, this results in five times more pretraining samples for sampled noise than for composed noise.

In subsequent epochs, we additionally consider $\mathcal{L}^{\text{back}}$ as training signal:

$$\mathcal{L}^{\text{back}} = \mathop{\mathbb{E}}_{x\sim\mathcal{G}}[-\log P_g(x|z^*(x))] + \mathop{\mathbb{E}}_{y\sim\mathcal{T}}[-\log P_t(y|w^*(y))]$$
$$z^*(x) = \arg\max_z P_t(z|x)$$
$$w^*(y) = \arg\max_w P_g(w|y)$$

This means that, in each iteration, we apply the current model to backtranslate a text (graph) to obtain a potentially imperfect graph (text) that we can use as noisy source with the clean original input being the target. This gives us a pseudo-parallel training instance for the next iteration – recall that we address unsupervised generation, i.e., without access to parallel data.

The total loss in these epochs is $\mathcal{L}^{\text{back}} + \mathcal{L}^{\text{denoise}}$, where now $\mathcal{L}^{\text{denoise}}$ only samples one possible type of noise independently for each data instance.

Supervised training. Our intended application is an unsupervised scenario. For our two datasets, however, we have labeled data (i.e., a "parallel corpus") and so can also compare our model to its supervised variant. Although supervised performance is generally better, it serves as a reference point and gives us an idea of the impact of supervision as opposed to factors like model architecture and hyperparameters. The supervised loss is simply defined as follows:

$$\mathcal{L}^{\text{sup}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{S}}[-\log P_t(y|x) - \log P_g(x|y)]$$

	VG	VG_{ball}	WebNLG
train split size	2,412,253	151,790	34,338
val split size	323,478	21,541	4,313
test split size	324,664	20,569	4,222
#relation types	36,506	5,167	373
avg #facts in graph	2.7	2.5	3.0
avg #tokens in text	5.4	5.5	22.8
avg % text tokens in graph	49.3	50.6	49.4
avg % graph tokens in text	52.3	54.7	75.6

Table 2: Statistics of WebNLG v2.1 and our newly created benchmark VG; VG_{ball} is a subset of VG representing images from ball sports events. Data split sizes are given as number of graph-text pairs.

5 Experiments

5.1 Data

For our experiments, we randomly split the VG images 80/10/10 into train/val/test. We then remove all graphs from train that also occur in one of the images in val or test. Finally, we unify graph serialization duplicates with different texts to single instances with multiple references for graph→text and proceed analogously with text duplicates for text→graph. For WebNLG v2.1, we use the data splits as provided. Following (Gardent et al., 2017a), we resolve the camel case of relation names and remove underscores from entity names in a preprocessing step. For both datasets, the order of facts in graph serializations corresponds to the order of triples in the original dataset. Because of VG's enormous size and limited computation power, we additionally create a closed-domain ball

graph → text	Visual Genome						WebNLG					
	BLEU		METEOR		CHRF++		BLEU		METEOR		CHRF++	
	val	test	val	test	val	test	val	test	val	test	val	test
$R^{graph \to text}$	5.9	5.9	28.2	28.1	43.4	43.3	18.3	18.3	33.5	33.6	55.0	55.2
Ours w/ sampled noise	19.8	19.5	31.4	31.2	50.9	50.7	**39.1**	**37.7**	**35.4**	**35.5**	**61.9**	**62.1**
Ours w/ composed noise	**23.2**	**23.2**	**33.0**	**32.9**	**53.7**	**53.6**	30.8	30.5	30.2	30.0	53.1	52.8
Ours *supervised*	26.5	26.4	32.3	32.2	53.7	53.6	35.1	34.4	39.6	39.5	64.1	64.0

Table 3: Results for unsupervised and supervised text generation. Note that training a supervised model on millions of labeled samples is usually not an option. Best unsupervised models are identified by best BLEU on $\mathcal{V}_{100}$. BLEU and METEOR are computed with scripts from (Lin et al., 2018); the CHRF++ script is from (Popović, 2017b).

sports subset of VG, called VG_{ball}, which we can use to quickly conduct additional experiments (see Section 7). We identify all images where at least one region graph contains at least one fact that mentions an object ending with *ball* and take all regions from them (keeping data splits the same). In contrast to alternatives like random subsampling, we consider this domain-focused construction more realistic.

Table 2 shows relevant statistics for all datasets. While VG and WebNLG have similar statistics, VG is around 70 times larger than WebNLG, which makes it an interesting benchmark for future research, both supervised and unsupervised. Apart from size, there are two important differences: (1) The VG graph schema has been freely defined by crowd workers and thus features a large variety of different relations. (2) The percentage of graph tokens occurring in the text, a measure important for the text→graph task, is lower for VG than for WebNLG. Thus, VG graphs contain more details than their corresponding texts, which is a characteristic feature of the domain of image captions: they mainly describe the salient image parts.

5.2 Training details

We train all models with the Adam optimizer (Kingma and Ba, 2015) for maximally 30 epochs. We stop supervised models early when $\mathcal{L}^{sup}$ does not decrease on val for 10 epochs. Unsupervised models are stopped after 5 iterations on VG because of its big size and limited computational resources. All hyperparameters and more details are described in Appendices A and B. Our implementation is based on AllenNLP (Gardner et al., 2017).

In unsupervised training, input graphs and texts are the same as in supervised training – only the gold target sides are ignored. While it is an artificial setup to split paired data and treat them as unpaired, this not only makes the supervised and unsupervised settings more directly comparable, but also ensures that the text data resemble the evaluation texts in style and domain. For the purpose of experiments on a benchmark, this seems appropriate to us. For a concrete use case, it would be an important first step to find adequate texts that showcase the desired language style and that are about a similar topic as the KGs that are to be textualized. As KGs are rarely the only means of storing information, e.g., in an industrial context, such texts should not be hard to come by in practice.

#	sampled noise				composed noise			
	$\mathcal{U}$	$\mathcal{V}_{100}$	val	test	$\mathcal{U}$	$\mathcal{V}_{100}$	val	test
1	**80.4**	7.8	10.1	9.9	**72.2**	15.9	19.8	19.7
2	50.7	7.2	9.2	9.1	41.2	14.0	15.2	15.1
3	67.6	19.5	19.4	19.2	61.0	22.7	**23.5**	**23.4**
4	56.4	**21.2**	**19.8**	**19.5**	51.9	22.2	21.4	21.3
5	62.9	20.0	19.6	19.4	60.5	**24.5**	23.2	23.2

Table 4: BLEU scores on VG for our unsupervised models evaluated for graph→text at different iterations. $\mathcal{U}$ is calculated on all unlabeled data used for training. $\mathcal{V}_{100}$ is a 100-size random sample from val. All results are computed with scripts from (Lin et al., 2018).

6 Results and Discussion

6.1 Text generation from graphs

Model selection. Table 4 shows how performance of our unsupervised model changes at every back-translation iteration, measured in BLEU (Papineni et al., 2002), a common metric for natural language generation. For model selection, we adopt the two methods proposed by Lample et al. (2018b), i.e., a small validation set (we take a 100-size random subset of val, called $\mathcal{V}_{100}$) and a fully unsupervised criterion ($\mathcal{U}$) where BLEU compares an unlabeled sample with its back-and-forth translation. We confirm their finding that $\mathcal{U}$ is not reliable for neural

(a) Reference text	a baseball cap on a baby's head
(b) $R^{graph \to text}$	baby is small and baby is wrapped in blanket and hat is pink and hat is baseball hat and baby wearing hat
(c) Unsuperv. neural model	small baby wrapped in blanket with pink baseball hat
(d) Superv. neural model	baby wearing a pink hat

Table 5: Texts generated from graph in Fig. 2.

	sampled noise				composed noise			
#	$\mathcal{U}$	$\mathcal{V}_{100}$	val	test	$\mathcal{U}$	$\mathcal{V}_{100}$	val	test
1	19.1	1.0	1.2	1.2	17.0	2.0	2.2	2.2
2	**71.0**	**21.7**	**19.1**	**18.8**	49.3	**22.1**	**22.1**	**21.7**
3	58.2	19.3	18.6	18.3	45.9	18.7	19.7	19.4
4	62.3	18.3	**19.1**	**18.8**	**54.4**	19.9	20.8	20.5
5	63.7	19.8	19.0	18.7	49.0	18.8	19.0	18.8

Table 6: F1 scores on VG for our models from Table 4 evaluated on text→graph at different iterations.

text → graph	VG		WebNLG	
	val	test	val	test
$R^{text \to graph}$	13.4	13.1	0.0	0.0
Stanford SG Parser	19.5	19.3	0.0	0.0
Ours w/ sampled noise	19.1	18.8	**38.5**	**39.1**
Ours w/ composed noise	**22.1**	**21.7**	32.5	33.1
Ours *supervised*	23.5	23.0	52.8	52.8

Table 7: F1 scores of facts extracted by our unsupervised semantic parsing (text→graph) systems and our model trained with supervision.

text generation models whereas $\mathcal{V}_{100}$ correlates better with performance on the larger test sets. We use $\mathcal{V}_{100}$ for model selection in the rest of this paper.

Quantitative evaluation. Table 3 shows BLEU, METEOR (Banerjee and Lavie, 2005) and CHRF++ (Popović, 2017a) for our unsupervised models and the rule baseline $R^{graph \to text}$, which is in many cases, i.e., if parallel graph-text data are scarce, the only alternative.

First, we observe that $R^{graph \to text}$ performs much better on WebNLG than VG, indicating that our new benchmark poses a tougher challenge. Second, our unsupervised models consistently outperform this baseline on all metrics and on both datasets, showing that our method produces textual descriptions much closer to human-generated ones. Third, noise composition, the general default in unsupervised machine translation, does not always perform better than noise sampling. Thus, it is worthwhile to try different noise settings for new tasks or datasets.

Surprisingly, supervised and unsupervised models perform nearly on par. Real supervision does not seem to give much better guidance in training than our unsupervised regime, as measured by our three metrics on two different datasets. Some metric-dataset combinations even favor one of the unsupervised models. Our qualitative observations provide a possible explanation for that.

Qualitative observations. Taking a look at example generations (Table 5), we also see qualitatively how much easier it is to grasp the content of our natural language summarization than reading through a simple enumeration of KG facts. We find that the unsupervised model (c) seems to output the KG information in a more complete manner than its supervised counterpart (d). The supervision probably introduces a bias present in the training data that image captions focus on salient image parts and therefore the supervised model is encouraged to omit information. As it never sees a corresponding text-graph pair together, the unsupervised model cannot draw such a conclusion.

6.2 Graph extraction from texts

We evaluate semantic parsing (text→graph) performance by computing the micro-averaged F1 score of extracted facts. If there are multiple reference graphs (cf. Section 5.1), an extracted fact is considered correct if it occurs in at least one reference graph. For the ground truth number of facts to be extracted from a given text, we take the maximum number of facts of all its reference graphs.

Model selection. Table 6 shows that (compared to text generation quality) $\mathcal{U}$ is more reliable for text→graph performance. For sampled noise, it correctly identifies the best iteration, whereas for composed noise it chooses second best. In both noise settings, $\mathcal{V}_{100}$ perfectly chooses the best model.

Quantitative observations. Table 7 shows a comparison of our unsupervised models with two rule-based systems, our $R^{text \to graph}$ and the highly domain-specific Stanford Scene Graph Parser (SSGP) by Schuster et al. (2015).

We choose these two baselines to adequately represent the state of the art in the unsupervised setting. Recall from Section 2 that the only previous unsupervised works either cannot adapt to a target graph schema (open information extraction), which means their precision and recall of retrieved facts is always 0, or have been created for SQL query

Input sentence	Man wearing a colorful shirt and white pants playing tennis
Reference (RG)	(shirt, attr, colorful) (pants, attr, white) (man, wearing, shirt) (man, wearing, pants)
$R^{text \to graph}$	(Man, wearing, colorful) (shirt, attr, colorful) (pants, attr, white) (pants, playing, tennis)
Stanford Scene Graph Parser	(shirt, play, tennis), (pants, play, tennis), (shirt, attr, colorful), (pants, attr, white)
Unsuperv. model w/ composed noise	(pants, attr, colorful) (pants, attr, white) (man, wearing, shirt) (man, playing, tennis)
Supervised model	(shirt, attr, colorful) (pants, attr, white) (Man, wearing, shirt) (Man, wearing, pants)

Table 8: Example fact extractions and evaluation wrt reference graph (RG). Green: correct ($\in$ RG). Yellow: acceptable fact, but $\notin$ RG. Red: incorrect ($\notin$ RG).

generation from natural language questions (Poon, 2013), a related task that is yet so different that an adaptation to triple set generation from natural language statements is nontrivial. While rule-based systems do not automatically adapt to new graph schemas either, $R^{text \to graph}$ and SSGP were at least designed with the scene graph domain in mind.

Although SSGP was not optimized to match the scene graphs from VG, its rules were still engineered to cover typical idiosyncrasies of textual image descriptions and corresponding scene graphs. Besides, we evaluate it with lemmatized reference graphs because it only predicts lemmata as predicates. All this gives it a major advantage over the other presented systems but it is nonetheless outperformed by our best unsupervised model – even on VG. This shows that our automatic method can beat even hand-crafted domain-specific rules.

Both $R^{text \to graph}$ and SSGP fail to predict any fact from WebNLG. The DBpedia facts from WebNLG often contain multi-token entities while $R^{text \to graph}$ only picks single tokens from the text. Likewise, SSGP models multi-token entities as two nodes

	VG_{ball}		WebNLG	
	g→t BLEU	t→g F1	g→t BLEU	t→g F1
No noise	0.9	0.0	14.8	0.0
sample all noise funs	**19.9**	17.3	**39.1**	**38.5**
compose all noise funs	19.6	**19.0**	30.8	32.5
use only `rule`	**19.5**	**18.5**	37.4	**31.0**
use only `swap`	0.9	0.0	13.1	0.0
use only `drop`	0.9	0.0	**39.9**	30.1
use only `blank`	0.9	0.0	14.9	0.0
use only `repeat`	1.1	0.0	15.7	0.0
sample all but `rule`	0.9	0.0	14.9	0.0
sample all but `swap`	19.2	17.0	39.6	**37.3**
sample all but `drop`	19.5	16.0	39.2	35.3
sample all but `blank`	19.9	**17.5**	**41.0**	37.0
sample all but `repeat`	**20.4**	16.6	36.7	37.1
comp. all but `rule`	0.9	0.0	13.5	0.0
comp. all but `swap`	20.2	16.3	35.9	40.8
comp. all but `drop`	21.5	18.6	36.4	41.1
comp. all but `blank`	20.2	16.3	34.8	40.4
comp. all but `repeat`	**21.1**	**20.1**	**38.5**	**42.3**

Table 9: Ablation study of our models on val of VG_{ball} and WebNLG v2.1. Models selected based on $\mathcal{V}_{100}$. Bold: best performance per column and block. Underlined: worse than corresponding rule-based system.

with an `attr` relation. This illustrates the importance of automatic adaptation to the target KG. Although our system uses $R^{text \to graph}$ during unsupervised training and is similarly not adapted to the WebNLG dataset, it performs significantly better.

Supervision helps more on WebNLG than on VG. The poor performance of $R^{text \to graph}$ on WebNLG is probably a handicap for unsupervised learning.

Qualitative observations. Table 8 shows example facts extracted by different systems. $R^{text \to graph}$ and SSGP are both fooled by the proximity of the noun *pants* and the verb *play* whereas our model correctly identifies *man* as the subject. It, however, fails to identify *shirt* as an entity and associates the two attributes *colorful* and *white* to *pants*. Only the supervised model produces perfect output.

6.3 Noise and translation completeness

Sampled noise only creates training pairs that either are complete rule-based translations or reconstruction pairs from a noisy graph to a complete graph or a noisy text to a complete text. In contrast, composed noise can introduce translations from a noisy text to a complete graph or vice versa and thus encourage a system to omit input information (cf. Fig. 3). This difference is mirrored nicely in the results of our unsupervised systems for both tasks: composed noise performs better on VG where omit-

ted information in an image caption is common and sampled noise works better on WebNLG where the texts describe their graphs completely.

7 Noise Ablation Study

Our unsupervised objectives are defined by different types of noise models. Hence, we examine their impact in a noise ablation study. Table 9 shows results for text→graph and graph→text on the validation splits of VG_{ball} and WebNLG.

For both datasets and tasks, introducing variation via noise functions is crucial for the success of unsupervised learning. The model without noise (i.e., $C(x) = x$) fails completely as do all models lacking `rule` as type of noise, the only exception being the only-`drop` system on WebNLG. Even though `drop` seems to work equally well in this one case, the simple translations delivered by our rule-based systems clearly provide the most useful information for the unsupervised models – notably in combination with the other noise functions: removing `rule` and keeping all other types of noise (cf. "sample all but `rule`" and "comp. all but `rule`") performs much worse than leaving out `drop`.

We hypothesize that our two rule systems provide two important pieces of information: (1) $R^{graph \to text}$ helps distinguish data format tokens from text tokens and (2) $R^{text \to graph}$ helps find probable candidate words in a text that form facts for the data output. As opposed to machine translation, where usually every word in a sentence is translated into a fluent sentence in the target language, identifying words that probably form a fact is more important in data-to/from-text generation.

We moreover observe that our unsupervised models always improve on the rule-based systems even when `rule` is the only type of noise: graph→text BLEU increases from 6.2/18.3 to 19.5/37.4 on VG_{ball}/WebNLG and text→graph F1 from 14.4/0.0 to 18.5/31.0.

Finally, our ablation study makes clear that there is no best noise model for all datasets and tasks. We therefore recommend experimenting with both different sets of noise functions and noise injection regimes (sampled vs. composed) for new data.

8 Conclusion

We presented the first fully unsupervised approach to text generation from KGs and a novel approach to unsupervised semantic parsing that automatically adapts to a target KG. We showed the effectiveness of our approach on two datasets, WebNLG v2.1 and a new text↔graph benchmark in the visual domain, derived from Visual Genome. We quantitatively and qualitatively analyzed our method on text↔graph conversion. We explored the impact of different unsupervised objectives in an ablation study and found that our newly introduced unsupervised objective using rule-based translations is essential for the success of unsupervised learning.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and gratefully acknowledge a Ph.D. scholarship awarded to the first author by the German Academic Scholarship Foundation (Studienstiftung des deutschen Volkes). This work was supported by by the BMBF as part of the project MLWin (01IS18050).

References

Mark van Assem, Aldo Gangemi, and Guus Schreiber. 2006. Conversion of wordnet to a standard rdf/owl representation. In *Proceedings of the Fifth Edition of the International Conference on Language Resources and Evaluation (LREC 2006)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computing Research Repository*, arXiv:1409.0473.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Rajarshi Bhowmik and Gerard de Melo. 2018. Generating fine-grained open vocabulary entity type descriptions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 877–888, Melbourne, Australia. Association for Computational Linguistics.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for AMR-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3649–3658, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4184–4194, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Keith Brown et al., editor, *Encyclopedia of Language and Linguistics*, second edition, pages 665–670. Elsevier, Oxford.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform. *Computing Research Repository*, arXiv:1803.07640.

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1486–1495, Portland, Oregon, USA. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *Computing Research Repository*, arXiv:1207.0580.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Aishwarya Kamath and Rajarshi Das. 2019. A survey on semantic parsing. In *Automated Knowledge Base Construction (AKBC)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, Minneapolis, Minnesota. Association for Computational Linguistics.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Computing Research Repository*, arXiv:1602.07332.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Tsung-Yi Lin, Xinlei Chen, Hao Fang, and Ramakrishna Vedantam. 2018. GitHub repository: tylin/coco-caption (Microsoft COCO caption evaluation). https://github.com/tylin/coco-caption.

Edward Loper and Steven Bird. 2004. Nltk: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2016. Visual relationship detection with language priors. In *European Conference on Computer Vision*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 1–9, Tilburg University, The Netherlands. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244.

Susan W. McRoy, Songsak Channarukul, and Syed S. Ali. 2000. YAG: A template-based generator for real-time systems. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 264–267, Mitzpe Ramon, Israel. Association for Computational Linguistics.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur P. Parikh, Hoifung Poon, and Kristina Toutanova. 2015. Grounded semantic parsing for complex knowledge extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 756–766, Denver, Colorado. Association for Computational Linguistics.

Hoifung Poon. 2013. Grounded unsupervised semantic parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 933–943, Sofia, Bulgaria. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore. Association for Computational Linguistics.

Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates. 2004. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 141–147, Geneva, Switzerland. COLING.

Maja Popović. 2017a. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Maja Popović. 2017b. GitHub repository: m-popovic/chrf (chrF). https://github.com/m-popovic/chrF.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-Text Generation with Content Selection and Planning. In *Proceedings of the 33rd Conference on Artificial Intelligence*.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3181–3192, Hong Kong, China. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs. *Computing Research Repository*, arXiv:2001.11003.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jacques Pierre Robin. 1995. *Revision-based Generation of Natural Language Summaries Providing Historical Background: Corpus-based Analysis, Design, Implementation and Evaluation*. Ph.D. thesis, Columbia University, New York, NY, USA. UMI Order No. GAX95-33653.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3027–3035.

Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D. Manning. 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80, Lisbon, Portugal. Association for Computational Linguistics.

Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 360–370, Tilburg University, The Netherlands. Association for Computational Linguistics.

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1378–1387, Florence, Italy. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 4444–4451. AAAI Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Kumiko Tanaka-Ishii, Koiti Hasida, and Itsuki Noda. 1998. Reactive content selection in the generation of real-time soccer commentary. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1282–1288, Montreal, Quebec, Canada. Association for Computational Linguistics.

Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1627–1637, Melbourne, Australia. Association for Computational Linguistics.

Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. 2008. *Handbook of knowledge representation*, volume 1. Elsevier.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Hai Wan, Yonghao Luo, Bo Peng, and Wei-Shi Zheng. 2018. Representation learning for scene graph completion via jointly structural and visual embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 949–956. International Joint Conferences on Artificial Intelligence Organization.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. 2018. DrugBank 5.0: a major update to the DrugBank database

for 2018. *Nucleic Acids Research*, 46(D1):D1074–D1082.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK. Association for Computational Linguistics.

A Hyperparameters

We use the following settings for all our experiments: learning rate of 10^{-4}, word embeddings of size 300, an LSTM hidden size of 250, a dropout rate of 0.2 and a batch size of 10. Following Lample et al. (2018b), we set $p_{\text{blank}} = p_{\text{repeat}} = 0.2$, $p_{\text{drop}} = 0.1$. For inference, we decode greedily with a maximum number of 40 decoding steps. To speed up unsupervised learning, we increase the batch size to 64 when creating backtranslations.

B Model details

We train with homogeneous batches of one target output type (text or graph) at a time. We use a single GeForce GTX 1080 GPU for training and inference. In this environment, pure training takes approximately 9 ms per instance and inference, which also means backtranslation, takes approximately 21 ms per instance. This means that unsupervised learning approximately needs 30 ms per instance. WebNLG models use 10.6 million parameters, VG models have 60.7 million parameters. The difference is due to a larger vocabulary size of 70,800 for VG compared to 8,171 for WebNLG.

C Results of all iterations on WebNLG

See Table 10 for all intermediate graph→text results of unsupervised training on WebNLG and Table 11 for text→graph. We find similar trends as for VG (Tables 4 and 6) except for $\mathcal{U}$ being a less reliable performance indicator for text→graph in the sampled noise setting.

	sampled noise			composed noise		
#	$\mathcal{U}$	$\mathcal{V}_{100}$	val	$\mathcal{U}$	$\mathcal{V}_{100}$	val
1	91.7	12.8	13.0	23.0	15.9	15.5
2	**94.0**	14.7	15.8	53.2	22.2	20.7
3	85.2	25.5	26.0	71.0	23.2	22.8
4	65.9	27.7	28.8	75.2	25.3	26.2
5	65.5	31.4	30.7	69.2	25.9	27.2
6	58.1	31.5	31.0	71.5	27.6	27.7
7	48.0	31.3	32.3	**79.2**	29.0	27.7
8	48.3	32.8	33.4	52.5	28.1	27.5
9	37.5	33.2	34.0	57.1	30.5	30.0
10	42.1	32.8	33.4	52.4	30.6	29.9
11	38.7	34.7	34.8	59.9	32.0	**31.6**
12	38.7	36.4	36.2	42.1	30.4	30.8
13	39.3	33.5	35.1	50.0	30.7	30.7
14	40.5	36.9	36.6	46.7	30.9	30.7
15	41.8	36.5	37.5	48.2	31.1	30.3
16	43.2	36.9	38.0	43.7	30.3	29.6
17	39.1	35.6	36.6	43.1	29.0	29.7
18	38.5	37.5	38.3	31.1	29.7	29.8
19	38.8	37.8	38.4	39.5	29.0	29.8
20	37.5	37.2	38.6	36.2	31.3	29.8
21	36.4	36.8	38.4	35.2	30.0	30.8
22	44.8	36.3	39.7	37.6	32.4	30.7
23	40.8	35.8	38.2	39.6	31.4	30.3
24	35.8	39.2	39.6	39.6	32.4	30.3
25	40.6	38.5	39.5	37.0	33.2	30.9
26	36.8	38.9	40.3	41.3	32.3	30.2
27	44.1	39.7	**40.6**	37.3	33.0	30.4
28	39.3	36.9	38.9	39.0	**34.7**	30.8
29	36.1	37.6	38.6	41.5	31.0	30.6
30	38.7	**40.7**	39.1	42.9	30.6	30.0

Table 10: BLEU scores on WebNLG for our unsupervised models evaluated for graph→text at different iterations. $\mathcal{U}$ is calculated on all unlabeled data used for training. $\mathcal{V}_{100}$ is a 100-size random sample from val. All results are computed with scripts from (Lin et al., 2018).

	sampled noise			composed noise		
#	$\mathcal{U}$	$\mathcal{V}_{100}$	val	$\mathcal{U}$	$\mathcal{V}_{100}$	val
1	**69.4**	0.0	0.0	0.0	0.0	0.0
2	64.0	0.0	0.1	16.2	1.2	1.6
3	35.6	0.9	0.3	7.5	3.3	3.0
4	47.8	2.6	2.3	37.5	5.5	5.5
5	39.2	5.7	3.4	35.3	7.0	6.6
6	39.2	6.2	5.6	44.9	9.7	8.0
7	45.8	9.8	7.9	58.3	8.0	10.3
8	50.0	12.6	10.0	51.1	14.0	12.8
9	54.9	13.6	12.9	53.1	12.5	14.0
10	58.3	14.9	14.3	51.1	15.9	16.8
11	62.5	19.3	17.8	53.8	15.6	17.3
12	54.2	20.3	18.2	58.3	16.7	18.0
13	57.1	23.1	20.2	47.8	19.8	20.6
14	37.5	25.5	21.4	49.0	20.6	22.1
15	48.0	25.7	22.4	54.2	23.0	22.8
16	52.0	27.9	24.3	46.2	22.5	25.4
17	50.0	26.7	25.1	35.6	26.8	26.8
18	48.0	32.1	27.7	52.2	27.8	27.7
19	56.0	32.3	28.9	58.3	26.4	28.1
20	60.0	31.0	30.1	55.3	26.4	29.2
21	51.0	32.3	30.4	59.3	27.6	30.7
22	55.3	34.9	32.0	**62.5**	31.7	32.0
23	44.9	34.3	32.7	54.9	34.0	32.6
24	58.8	38.4	33.7	61.2	31.5	32.4
25	46.8	39.6	34.1	58.3	33.3	33.1
26	53.8	40.6	36.3	54.2	**34.4**	32.5
27	62.5	41.8	36.4	50.0	33.9	33.3
28	55.3	41.0	37.4	40.8	32.6	**33.7**
29	56.0	40.7	37.0	58.8	29.5	**33.7**
30	59.6	**41.9**	**38.5**	53.8	31.6	33.4

Table 11: F1 scores on WebNLG for our unsupervised models evaluated for text→graph at different iterations. $\mathcal{U}$ is calculated on all unlabeled data used for training. $\mathcal{V}_{100}$ is a 100-size random sample from val.

7 Conclusion

In this thesis, we presented advances in the area of artificial intelligence, in particular, in the tasks of scene graph classification through modeling and utilization of commonsense knowledge. In the following, we summarize our primary contributions and outline promising future research directions.

- In Chapter 2, we introduced image-grounded symbol representations called Schemata. We proposed a deep learning-based architecture that could directly learn symbol representations from very few images using a self-supervised backbone. We showed that top-down injection of schema representations to images during test time could largely improve scene graph classification. Furthermore, this framework enabled us to design a novel approach for fine-tuning the perception pipeline from external knowledge graphs instead of annotated images. We showed that in this way, we could achieve similar accuracy in predicate classification, using 1% of the annotated images, while obtaining competitive results in object classification and scene graph classification.

- In Chapter 3, we proposed a transformer-based architecture that could exploit the relational knowledge in texts and improve the scene graph classification. We used a pre-trained T5 model that could convert any unstructured text to structured knowledge graphs. We then mapped the extracted knowledge to pre-trained class prototypes (schemata) and fine-tuned a graph transformer module in our perception pipeline. We showed that this gives us 8x better scene graph classification, 3x better object classification, and 1.5x better predicate classification compared to supervised baselines with 1% of the annotated images.

- In Chapter 4, we argued that current methods for visual relation detection cannot distinguish between some relations given the lack of 3D information. Therefore, we created a dataset, *VG-Depth*, consisting of the predicted depth maps for more than 60k images from Visual Genome [Krishna et al., 2017]. We introduced a

convolutional neural network-based fusion method that, for the first time, utilizes both the RGB *and* Depth images to solve the visual relation detection task. Through extensive experiments using different features, we showed that using depth maps in visual relation detection significantly improves state-of-the-art results in this domain.

- In Chapter 5, we proposed a biologically plausible, computational cognitive model that captures the connection between perception, semantic memory, and episodic memory. To evaluate our model in a scene understanding scenario, we performed experiments on the Stanford Visual Relation Detection (VRD) dataset and showed that semantic memories could be learned directly from perception.

- In Chapter 6, we show that there is no need for parallel text-graph data to train models for the tasks of text-to-graph and graph-to-text. Instead, we proposed an unsupervised architecture inspired by back-translation. We showed that our models outperform strong baselines for both tasks on WebNLG and our manually created Visual Genome-based dataset without any manual adaptation between datasets.

In summary, we made contributions towards improving scene understanding using top-down utilization of commonsense, either in the form of structured (Chapter 2) or unstructured (Chapter 3, Chapter 6) visual-relational knowledge, or 3D (Chapter 4) knowledge. We showed the biological and cognitive plausibility of our approaches (Chapter 5).

Our publications and openly available code repositories make us confident that these research directions will be successfully pursued in the future. In particular, in the future, it will be interesting to explore further the capabilities of large visual language models in modeling, utilizing, and manipulating structured knowledge as well as extending to tasks such as visual reasoning and visual question answering.

Bibliography

Sahand Sharifzadeh, Sina Moayed Baharlou, and Volker Tresp. Classification by attention: Scene graph classification with prior knowledge. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, pages 5025–5033, Online, 2021.

Sahand Sharifzadeh, Sina Moayed Baharlou, Martin Schmitt, Hinrich Schütze, and Volker Tresp. Improving scene graph classification by exploiting knowledge from texts. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-22)*, Online, 2022.

Sahand Sharifzadeh, Sina Moayed Baharlou, Max Berrendorf, Rajat Koner, and Volker Tresp. Improving visual relation detection using depth maps. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3597–3604. IEEE, 2020.

Volker Tresp, Sahand Sharifzadeh, and Dario Konopatzki. A model for perception and memory. *Conference on Cognitive Computational Neuroscience*, 2019.

Martin Schmitt, Sahand Sharifzadeh, Volker Tresp, and Hinrich Schütze. An unsupervised joint system for text generation from knowledge graphs and semantic parsing. *arXiv preprint arXiv:1904.09447*, 2019.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020b.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

Jeffrey S Johnson and Bruno A Olshausen. The recognition of partially visible natural objects in the presence and absence of their occluders. *Vision research*, 45(25-26): 3262–3276, 2005.

Dean Wyatte, David J Jilk, and Randall C O'Reilly. Early recurrent feedback facilitates visual object recognition under challenging conditions. *Frontiers in psychology*, 5:674, 2014.

Hanlin Tang, Calin Buia, Radhika Madhavan, Nathan E Crone, Joseph R Madsen, William S Anderson, and Gabriel Kreiman. Spatiotemporal dynamics underlying object completion in human ventral visual cortex. *Neuron*, 83(3):736–748, 2014.

Courtney J Spoerer, Patrick McClure, and Nikolaus Kriegeskorte. Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in psychology*, 8:1551, 2017.

Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

Olaf Sporns and Jonathan D Zwi. The small world of the cerebral cortex. *Neuroinformatics*, 2(2):145–162, 2004.

Nikola T Markov, Julien Vezoli, Pascal Chameau, Arnaud Falchier, René Quilodran, Cyril Huissoud, Camille Lamy, Pierre Misery, Pascale Giroud, Shimon Ullman, et al. Anatomy of hierarchy: feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1):225–259, 2014.

Volker Tresp, Sahand Sharifzadeh, Dario Konopatzki, and Yunpu Ma. The tensor brain: Semantic decoding for perception and memory. *arXiv preprint arXiv:2001.11027*, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016.

Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42 (1-3):335–346, 1990.

David Pitt. Mental Representation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2020 edition, 2020.

Immanuel Kant. *Kritik der reinen Vernunft:[Hauptband]*. Walter de Gruyter, 1787.

Jean Piaget. *Langage et pensée chez l'enfant*. Delachaux et Niestlé, 1923.

Michael A Arbib. Schema theory. *The encyclopedia of artificial intelligence*, 2:1427–1443, 1992.

Vyvyan Evans. *Cognitive linguistics*. Edinburgh University Press, 2006.

Bernard St-Louis, Marieve Corbeil, Andre Achim, and Stevan Harnad. Acquiring the mental lexicon through sensorimotor category learning. 2008.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, Georgia, USA, 2013.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27, 2014.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European conference on computer vision*, pages 852–869. Springer, 2016.

Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11546–11556, 2021.

Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *arXiv preprint arXiv:2201.11460*, 2022.

Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, et al. Relationformer: A unified framework for image-to-graph generation. *arXiv preprint arXiv:2203.10202*, 2022.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2019.

Sahand Sharifzadeh, Ioannis Chiotellis, Rudolph Triebel, and Daniel Cremers. Learning to drive using inverse reinforcement learning and deep q-networks. *arXiv preprint arXiv:1612.03653*, 2016.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1): 11–33, 2016.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 809–816, 2011.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014.

Mehdi Ali, Max Berrendorf*, Charles Tapley Hoyt*, Laurent Vermue*, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *CoRR*, abs/2006.13365, 2020. * equal contribution.

Mehdi Ali, Max Berrendorf*, Charles Tapley Hoyt*, Laurent Vermue*, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research*, 22(82): 1–6, 2021. URL `http://jmlr.org/papers/v22/20-825.html`. * equal contribution.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.