
Multimodal Deep Learning



Contents

Preface	v
Foreword	1
1 Introduction	3
1.1 Introduction to Multimodal Deep Learning	3
1.2 Outline of the Booklet	4
2 Introducing the modalities	7
2.1 State-of-the-art in NLP	10
2.2 State-of-the-art in Computer Vision	10
2.3 Resources and Benchmarks for NLP, CV and multimodal tasks	30
3 Multimodal architectures	61
3.1 Img2text	65
3.2 Text-2-image	78
3.3 Images supporting language models	81
3.4 Text supporting computer vision models	115
3.5 Text + Image	128
4 Further Topics	149
4.1 Including Further Modalities	149
4.2 Structured + Unstructured Data	153
4.3 Multi-Purpose Models	166
4.4 Generative Art	181
5 Conclusion	189

6 Epilogue	191
7 Acknowledgements	193

Preface

In the last few years, there have been several breakthroughs in the methodologies used in Natural Language Processing (NLP) as well as Computer Vision (CV). Beyond these improvements on single-modality models, large-scale multi-modal approaches have become a very active area of research.

In this seminar, we reviewed these approaches and attempted to create a solid overview of the field, starting with the current state-of-the-art approaches in the two subfields of Deep Learning individually. Further, modeling frameworks are discussed where one modality is transformed into the other Chapter 3.1 and Chapter 3.2), as well as models in which one modality is utilized to enhance representation learning for the other (Chapter 3.3 and Chapter 3.4). To conclude the second part, architectures with a focus on handling both modalities simultaneously are introduced (Chapter 3.5). Finally, we also cover other modalities (Chapter 4.1 and Chapter 4.2) as well as general-purpose multi-modal models (Chapter 4.3), which are able to handle different tasks on different modalities within one unified architecture. One interesting application (Generative Art, Chapter 4.4) eventually caps off this booklet.



FIGURE 1: Creative Commons License

This book is licensed under the [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#).



Foreword

Author: Matthias Aßnenmacher

This book is the result of an experiment in university teaching. We were inspired by a group of other PhD Students around Christoph Molnar, who conducted another [seminar on Interpretable Machine Learning](#) in this format. Instead of letting every student work on a seminar paper, which more or less isolated from the other students, we wanted to foster collaboration between the students and enable them to produce a tangible output (that isn't written to spend the rest of its time in (digital) drawers). In the summer term 2022, some Statistics, Data Science and Computer Science students signed up for our seminar entitled "Multimodal Deep Learning" and had (before kick-off meeting) no idea what they had signed up for: Having written an entire book by the end of the semester.

We were bound by the examination rules for conducting the seminar, but otherwise we could deviate from the traditional format. We deviated in several ways:

1. Each student project is a chapter of this booklet, linked contentwise to other chapters since there's partly a large overlap between the topics.
 2. We gave challenges to the students, instead of papers. The challenge was to investigate a specific impactful recent model or method from the field of NLP, Computer Vision or Multimodal Learning.
 3. We designed the work to live beyond the seminar.
 4. We emphasized collaboration. Students wrote the introduction to chapters in teams and reviewed each others individual texts.
-

Citation

If you refer to the book, please use the following citation (authors in alphabetical order):

`@misc{seminar_22_multimodal,`

```
title = {Multimodal Deep Learning},  
author = {Akkus, Cem and Chu, Luyang and Djakovic, Vladana and  
Jauch-Walser, Steffen and Koch, Philipp and Loss,  
Giacomo and Marquardt, Christopher and Moldovan,  
Marco and Sauter, Nadja and Schneider, Maximilian  
and Schulte, Ricker and Urbanczyk, Karol and  
Goschenhofer, Jann and Heumann, Christian and  
Hvingelby, Rasmus and Schalk, Daniel and  
Aßenmacher, Matthias},  
url = {https://slds-lmu.github.io/seminar_multimodal_dl/},  
day = {30},  
month = {Sep},  
year = {2022}  
}
```

Technical Setup

The book chapters are written in the Markdown language. The simulations, data examples and visualizations were created with R ([R Core Team, XXXX](#)). To combine R-code and Markdown, we used rmarkdown. The book was compiled with the bookdown package. We collaborated using git and github. For details, head over to the [book's repository](#).

1

Introduction

Author: Nadja Sauter

Supervisor: Matthias Assenmacher

1.1 Introduction to Multimodal Deep Learning

There are five basic human senses: hearing, touch, smell, taste and sight. Processing these five modalities, we are able to perceive and understand the world around us. Thus, multimodal means to combine different channels of information simultaneously to understand our surroundings. For example, when toddlers learn the word “cat”, they use different modalities by saying the word out loud, pointing on cats and making sounds like “meow”. Using the human learning process as a role model, AI researchers also try to combine different modalities to train deep learning models. On a superficial level, deep learning algorithms are based on a neural network that is trained to optimize some objective which is mathematically defined via the so-called loss function. The optimization of minimizing the loss is done by a mathematical operation called gradient descent. Consequently, deep learning models can only handle numeric input and can only result in a numeric output. However, in multimodal tasks we are often confronted with unstructured data like pictures or text. Thus, the first major problem is how to represent the input numerically. The second issue with regard to multimodal tasks is how exactly to combine different modalities. For instance, a typical task could be to train a deep learning model to generate a picture of a cat. First of all, the computer needs to understand the text input “cat” and then somehow translate this information into a specific image. Therefore, it is necessary to identify the contextual relationships between words in the text input and the spatial pixel relationships in the image output. What might be easy for a toddler in pre-school, is a huge challenge for the computer. Both have to learn some understanding of the word “cat” that comprises the meaning and appearance of the animal. A common approach in artificial intelligence (AI) is to generate a high-level embedding that represents the cat numerically as a vector in some latent space.

However, to achieve this, different approaches and algorithmic architectures have been developed in recent years. This book gives an overview of the different methods used in state-of-the-art multimodal deep learning to overcome challenges arising with unstructured data and combining inputs of different modalities.

1.2 Outline of the Booklet

Since multimodal models often use text and images as input or output, methods of Natural Language Processing (NLP) and Computer Vision (CV) are introduced as foundation in Chapter 2. Methods in the area of NLP try to handle text data, whereas CV deals with image processing. With regard to NLP (subsection 2.1), one concept of major importance is the so-called word embedding, which is nowadays an essential part of all multimodal deep learning architectures. This concept also sets the foundation for transformer-based models like BERT (Devlin et al., XXXXa), which achieved a huge improvement in several NLP tasks. Especially the attention mechanism (Vaswani et al., XXXXa) of transformers revolutionized NLP models which is why most of them rely on the transformer as a backbone. In Computer Vision (subsection 2.2) different network architectures, namely ResNet (He et al., XXXXb), EfficientNet (Tan and Le, XXXX), SimCLR (Chen et al., XXXXa) and BYOL (Grill et al., XXXXb), will be introduced. Here it is of great interest to compare the different approaches in CV and their performance on challenging benchmarks. For this reason, the last subsection 2.3 of chapter 1 gives an overall overview of different data sets, pre-training tasks and benchmarks for CV as well as for NLP.

The second Chapter (see 3) focuses on different multimodal architectures, covering a wide variety of how text and images can be combined. The presented models combine and advance different methods of NLP and CV. First of all, looking at Img2Text tasks (subsection 3.1), the data set Microsoft COCO for object recognition (Lin et al., XXXXa) and the meshed-memory transformer for Image Captioning (M^2 Transformer) (Cornia et al., XXXX) will be presented. Contrariwise, researchers developed methods to generate pictures based on a short text prompt (subsection 3.2). The first models accomplishing this task were generative adversarial networks (GANs) (Goodfellow et al., XXXXb) and Variational Autoencoders (VAEs) (Kingma and Welling, XXXX). These methods were improved in recent year and today state-of-the art transformer architectures with text-guided diffusion models like DALL-E (Ramesh et al., XXXXb) and GLIDE (Nichol et al., XXXX) achieve remarkable results. Another interesting question is how images can be utilized to support language models (subsection 3.3). This can be done via sequential

embeddings, more advanced grounded embeddings or, again, inside transformers. On the other hand, you can also look at text supporting CV models like CLIP (Radford et al., XXXXb), ALIGN (Jia et al., XXXXa) and Florence (Yuan et al., XXXXa) (subsection 3.4). They use foundation models meaning reusing models (e.g. CLIP inside DALL-E 2) as well as a contrastive loss for connecting text with images. Besides, zero-shooting makes it possible to classify new and unseen data. Especially the open-source architecture CLIP (Radford et al., XXXXb) for image classification and generation attracted a lot of attention last year. In the end of the second chapter, some further architectures to handle text and images simultaneously are introduced (subsection 3.5). For instance, Data2Vec uses the same learning method for speech, vision and language and in this way aims to find a general approach to handle different modalities in the same way. Furthermore, VilBert (Lu et al., XXXXa) extends the popular BERT architecture to handle both image and text as input by implementing co-attention. This method is also used in Google's Deepmind Flamingo (Alayrac et al., XXXXb). In addition, Flamingo aims to tackle multiple tasks with a single visual language model via few-shot learning and freezing the pre-trained vision and language model.

In the last chapter (see 4), methods are introduced that are also able to handle modalities other than text and image, like e.g. video or speech. The overall goal here is to find a general multimodal architecture based on challenges rather than modalities. Therefore, one needs to handle problems of multimodal fusion and alignment and decide whether you use a join or coordinated representation (subsection 4.1). Moreover we go more into detail about how exactly to combine structured and unstructured data (subsection 4.2). Therefore, different fusion strategies which evolved in recent years will be presented. This is illustrated in this book by two use cases in survival analysis and economics. Besides this, another interesting research question is how to tackle different tasks in one so called multi-purpose model (subsection 4.3) like it is intended to be done by Google researchers Barham et al. (XXXX) and their Pahtway model. Last but not least, we show one exemplary application of Multimodal Deep Learning in the arts scene where image generation models like DALL-E (Ramesh et al., XXXXb) are used to create art pieces in the area of Generative Arts (subsection 4.4).



2

Introducing the modalities

Authors: Cem Akkus, Vladana Djakovic, Christopher Benjamin Marquardt

Supervisor: Dr. Matthias Aßenmacher

Natural Language Processing (NLP) has existed for about 50 years, but it is more relevant than ever. There have been several breakthroughs in this branch of machine learning that is concerned with spoken and written language. For example, learning internal representations of words was one of the greater advances of the last decade. Word embeddings (?, ?) made it possible and allowed developers to encode words as dense vectors that capture their underlying semantic content. In this way, similar words are embedded close to each other in a lower-dimensional feature space. Another important challenge was solved by Encoder-decoder (also called sequence-to-sequence) architectures ?, which made it possible to map input sequences to output sequences of different lengths. They are especially useful for complex tasks like machine translation, video captioning or question answering. This approach makes minimal assumptions on the sequence structure and can deal with different word orders and active, as well as passive voice.

A definitely significant state-of-the-art technique is Attention ?, which enables models to actively shift their focus – just like humans do. It allows following one thought at a time while suppressing information irrelevant to the task. As a consequence, it has been shown to significantly improve performance for tasks like machine translation. By giving the decoder access to directly look at the source, the bottleneck is avoided and at the same time, it provides a shortcut to faraway states and thus helps with the vanishing gradient problem. One of the most recent sequence data modeling techniques is Transformers ([Vaswani et al. \(XXXXb\)](#)), which are solely based on attention and do not have to process the input data sequentially (like RNNs). Therefore, the deep learning model is better in remembering context-induced earlier in long sequences. It is the dominant paradigm in NLP currently and even makes better use of GPUs, because it can perform parallel operations. Transformer architectures like BERT ([Devlin et al. \(XXXXb\)](#)), T5 (?) or GPT-3 ([Brown et al. \(XXXXa\)](#)) are pre-trained on a large corpus and can be fine-tuned for specific language tasks. They have the capability to generate stories, poems, code and much more. With the help of the aforementioned breakthroughs, deep networks have been successful in retrieving information and finding rep-

resentations of semantics in the modality text. In the next paragraphs, developments for another modality image are going to be presented.

Computer vision (CV) focuses on replicating parts of the complexity of the human visual system and enabling computers to identify and process objects in images and videos in the same way that humans do. In recent years it has become one of the main and widely applied fields of computer science. However, there are still problems that are current research topics, whose solutions depend on the research's view on the topic. One of the problems is how to optimize deep convolutional neural networks for image classification. The accuracy of classification depends on width, depth and image resolution. One way to address the degradation of training accuracy is by introducing a deep residual learning framework ([He et al., XXXXb](#)). On the other hand, another less common method is to scale up ConvNets, to achieve better accuracy is by scaling up image resolution. Based on this observation, there was proposed a simple yet effective compound scaling method, called EfficientNets ([Tan and Le, XXXX](#)).

Another state-of-the-art trend in computer vision is learning effective visual representations without human supervision. Discriminative approaches based on contrastive learning in the latent space have recently shown great promise, achieving state-of-the-art results, but the simple framework for contrastive learning of visual representations, which is called SimCLR, outperforms previous work ([Chen et al., XXXXa](#)). However, another research proposes as an alternative a simple “swapped” prediction problem where we predict the code of a view from the representation of another view. Where features are learned by Swapping Assignments between multiple Views of the same image (SwAV) (?). Further recent contrastive methods are trained by reducing the distance between representations of different augmented views of the same image ('positive pairs') and increasing the distance between representations of augmented views from different images ('negative pairs'). Bootstrap Your Own Latent (BYOL) is a new algorithm for self-supervised learning of image representatiois ([Grill et al., XXXXb](#)).

Self-attention-based architectures, in particular, Transformers have become the model of choice in natural language processing (NLP). Inspired by NLP successes, multiple works try combining CNN-like architectures with self-attention, some replacing the convolutions entirely. The latter models, while theoretically efficient, have not yet been scaled effectively on modern hardware accelerators due to the use of specialized attention patterns. Inspired by the Transformer scaling successes in NLP, one of the experiments is applying a standard Transformer directly to the image (?). Due to the widespread application of computer vision, these problems differ and are constantly being at the center of attention of more and more research.

With the rapid development in NLP and CV in recent years, it was just a question of time to merge both modalities to tackle multi-modal tasks. The

release of DALL-E 2 just hints at what one can expect from this merge in the future. DALL-E 2 is able to create photorealistic images or even art from any given text input. So it takes the information of one modality and turns it into another modality. It needs multi-modal datasets to make this possible, which are still relatively rare. This shows the importance of available data and the ability to use it even more. Nevertheless, all modalities are in need of huge datasets to pre-train their models. It's common to pre-train a model and fine-tune it afterwards for a specific task on another dataset. For example, every state-of-the-art CV model uses a classifier pre-trained on an ImageNet based dataset. The cardinality of the datasets used for CV is immense, but the datasets used for NLP are of a completely different magnitude. BERT uses the English Wikipedia and the Books corpus to pre-train the model. The latter consists of almost 1 billion words and 74 million sentences. The pre-training of GPT-3 is composed of five huge corpora: CommonCrawl, Books1 and Books2, Wikipedia and WebText2. Unlike language model pre-training that can leverage tremendous natural language data, vision-language tasks require high-quality image descriptions that are hard to obtain for free. Widely used pre-training datasets for VL-PTM are Microsoft Common Objects in Context (COCO), Visual Genome (VG), Conceptual Captions (CC), Flickr30k, LAION-400M and LAION-5B, which is now the biggest openly accessible image-text dataset.

Besides the importance of pre-training data, there must also be a way to test or compare the different models. A reasonable approach is to compare the performance on specific tasks, which is called benchmarking. A nice feature of benchmarks is that they allow us to compare the models to a human baseline. Different metrics are used to compare the performance of the models. Accuracy is widely used, but there are also some others. For CV the most common benchmark datasets are ImageNet, ImageNetReaL, CIFAR-10(0), OXFORD-IIIT PET, OXFORD Flower 102, COCO and Visual Task Adaptation Benchmark (VTAB). The most common benchmarks for NLP are General Language Understanding Evaluation (GLUE), SuperGLUE, SQuAD 1.1, SQuAD 2.0, SWAG, RACE, ReCoRD, and CoNLL-2003. VTAB, GLUE and SuperGLUE also provide a public leader board. Cross-modal tasks such as Visual Question Answering (VQA), Visual Commonsense Reasoning (VCR), Natural Language Visual Reasoning (NLVR), Flickr30K, COCO and Visual Entailment are common benchmarks for VL-PTM.

2.1 State-of-the-art in NLP

2.2 State-of-the-art in Computer Vision

Author: Vladana Djakovic

Supervisor: Daniel Schalk

2.2.1 History

The first research about visual perception comes from neurophysiological research performed in the 1950s and 1960s on cats, where they used cats as a model to understand how human vision is compounded. Scientists concluded that human vision is hierarchical, and neurons detect simple features like edges, followed by more complex features like shapes and more complex visual representations. Inspired by this knowledge, computer scientists focused on recreating human neurological structures.

At around the same time, as computers became more advanced, computer scientists worked on imitating human neurons' behavior and simulating a hypothetical neural network. In his book "The Organization of Behaviour" (1949) Donald Hebb stated that neural pathways strengthen over each successive use, especially between neurons that tend to fire at the same time, thus beginning the long journey towards quantifying the complex processes of the brain. The first Hebbian network, inspired by this neurological research, was successfully implemented at MIT in 1954 ?.

New findings led to the establishment of the field of artificial intelligence in 1956 on-campus at Dartmouth College. Scientists began to develop ideas and research how to create techniques that would imitate the human eye.

In 1959 early research on developing neural networks was performed at Stanford University, where models called "ADALINE" and "MADALINE," Multiple ADAptive LINear Elements, were developed. Those models aimed to recognize binary patterns and could predict the next bit ?.

Starting optimism about Computer Vision and neural networks disappeared after 1969 and the publication of the book "Perceptrons" by Marvin Minsky, founder of the MIT AI Lab. According to the authors of this book, the single perception approach to neural networks could not be translated effectively into multi-layered neural networks. The period that followed was known as AI Winter, which lasted until 2010, when the technological development of

computer and the internet became widely used. In 2012 breakthroughs in Computer Vision happened at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The team from the University of Toronto issued a deep neural network called AlexNet [7] that changed the field of artificial intelligent Computer Vision (CV). AlexNet achieved an error rate of 16.4%.

From then until today, Computer Vision has been one of the fastest developing fields. Researchers are competing to develop a model that would be the most similar to the human eye and help humans in everyday life. In this work the author will describe only a few recent state-of-the-art models.

2.2.2 Supervised and unsupervised learning

As part of artificial intelligence (AI) and machine learning (ML), there are two basic approaches:

- supervised learning;
- unsupervised learning.

Supervised learning [7] is used to train algorithms on labeled datasets that accurately classify data or predict outcomes. With labeled inputs and outputs model can measure its accuracy and learn over time. We can distinguish two types of data mining problems:

- classification,
- regression.

In unsupervised learning [7], unlabelled datasets are analyzed and clustered using machine learning algorithms. These algorithms aim to discover hidden patterns or data groupings without previous human intervention. The ability to find similarities and differences in information is mainly used for three main tasks:

- clustering,
- association,
- dimensionality reduction.

Solving the problems where the dataset can be both labeled and unlabeled requires a semi-supervised approach that lies between supervised and unsupervised learning. It is useful when extracting relevant features from data that is complex and when data is high volume, i.e., medical images. Nowadays, a new research topic appeared in the machine learning community, and it is Self-Supervised Learning. Self-Supervised learning is a process where the

model trains itself to learn one part of the input from another ?. As a subset of unsupervised learning, it involves machines labeling, categorizing, and analyzing information independently and drawing conclusions based on connections and correlations. It can also be considered an autonomous form of supervised learning since it does not require human input to label data. Unlike unsupervised learning, self-supervised learning does not focus on clustering nor grouping ?. One part of Self-Supervised learning is contrastive learning, which is used to learn the general features of an unlabeled dataset identifying similar and dissimilar data points. It is utilized to train the model to learn about our data without any annotations or labels ?.

2.2.3 Scaling networks

Ever since the introduction of AlexNet in 2012, the problem of scaling convolutional neural networks has become the topic of active research. ConvNet can be scaled in all three dimensions: depth, width, or image size. One of the first researches in 2015 showed that network depth is crucial for image classification. The question whether stacking more layers enables the network to learn better leads He K., Zhang X., et al. to deep residual networks called ResNet ([He et al., XXXB](#)), which will be described in this work. Later on, scaling networks by their depth became the most popular way to improve their performance. The second solution was to scale ConvNets by their width. Wider networks tend to be able to capture more fine-grained features and are easier to train ?. Lastly, scaling the image's resolution can improve the network's performance. With higher resolution input images, ConvNets could capture more fine-grained patterns. GPipe is one of the most famous networks created by this technique ?. The question of possibility of scaling by all three dimensions was answered by Tan M. et al. in 2019 in the work presenting EfficientNet. This network was built by scaling up ConvNets by all three dimensions and will also be described here.

2.2.4 Deep residual networks

The deep residual networks, called ResNets, (?) were presented as the answer on the question whether stacking more layers would enable network to learn better. Until then one obstacle for simply stacking layers was the problem of vanishing/exploding gradients. It has been primarily addressed by normalized initialization and intermediate normalization layers. That enabled networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation.

Another obstacle was a degradation problem. It occurs when the network depth increases, followed by saturating and then rapidly decreasing accuracy. Overfitting is not caused by such degradation, and adding more layers to a

suitably deep model leads to higher training error, which indicates that not all systems are similarly easy to optimize.

For example, it was suggested to consider a shallower architecture and its deeper counterpart that adds more layers. One way to avoid the degradation problem is to create a deeper model, where the auxiliary layers are identity mappings, and other layers are copied from a shallower model. The deeper model should produce no higher training error than its shallower counterpart. However, in practice, it is not the case, and it is hard to find comparably good to construct or better solutions. The solution to this degradation problem proposed by them is a deep residual learning framework.

2.2.4.1 Deep Residual Learning

2.2.4.1.1 Residual Learning

The idea of residual learning is to replace the approximation of underlying mapping $H(x)$, which is approximated by a few stacked layers (not necessarily the entire net), with an approximation of residual function $F(x) := H(x) - x$. Here x denotes the inputs to the first of these layers, and it is assumed that both inputs and outputs have the same dimensions. The original function change its form $F(x) + x$.

A counterintuitive phenomenon about degradation motivated this reformulation. The new deeper model should have no more significant training error when added layers are constructed as identity mappings. However, due to degradation problem, solvers may have challenges approximating identity mappings by multiple nonlinear layers. Using the residual learning reformulation, can drive the weights of the nonlinear layers toward zero to approach identity mappings if they are optimal. Generally, identity mappings are not optimal, but new reformulations may help to precondition the problem. When an optimal function is closer to an identity mapping than a zero mapping, finding perturbations concerning an identity mapping should be easier than learning the function from scratch.

2.2.4.1.2 Identity Mapping by Shortcuts

Residual learning is adopted to every few stacked layers where a building block is defined as shown in Fig. 1:

$$y = F(x, \{W_i\}) + x \quad (1)$$

x and y present the input and output vectors of the layers.

The function $F(x, \{W_i\})$ represents the residual mapping that is to be learned. For the example with two layers from Fig.1 that has two layers,

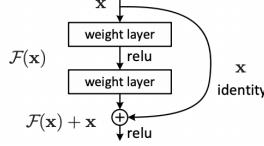


FIGURE 2.1: Figure 1. Building block of residual learning

$F = W_2\sigma(W_1x)$ in which σ denotes the ReLU activation function, and to simplify the notations, biases are left out. The operation $F + x$ is conducted with a shortcut connection and element-wise addition. Afterward, He et al. applied second nonlinearity (i.e., $\sigma(y)$, Fig.1).

The shortcut connections in Eq. (1) neither adds an extra parameter nor increases computation complexity, which enables comparisons between plain and residual networks that concurrently have the same number of parameters, depth, width, and computational cost (except for the negligible element-wise addition). Dimensions of x and F must be equal in Eq. (1). Alternatively, to match the dimensions, linear projection W_s by the shortcut connections can be applied:

$$y = F(x, \{W_i\}) + W_s x. \quad (2)$$

The square matrix W_s can be used in Eq. (1). However, experiments showed that identity mapping is enough to solve the degradation problem. Therefore, W_s only aims to match dimensions. Although more levels are possible, it was experimented with function F having two or three layers without stating the exact form of it. Assuming F only has one layer, Eq (1) it is comparable to a linear layer: $y = W_1x + x$. The theoretical notations are about fully-connected layers, but convolutional layers were used. The function $F(x, \{W_i\})$ can be applied to represent multiple convolutional layers. Two feature maps are added element-wise, channel by channel.

2.2.4.2 Network Architectures

To construct efficient residual network, various plain/residual networks were tested. They trained the network on benchmarked datasets, one of them being the ImageNet dataset, which will be used for a comparison of network architectures. Fig. 2 shows that every residual network needs a plain baseline network inspired by the VGG network ? on which identity mapping by shortcuts is applied.

Plain Network The philosophy of VGG nets 41 mainly inspires plain baselines. There are two rules that convolution layers, which usually have 3x3 filters, follow:

- feature maps with the same output size have the same number of layers;
- reducing the size of a feature map by half doubles the number of filters per layer to maintain time complexity per layer

Convolutional layers with a stride of 2 perform downsampling directly. A global average pooling layer and a 1000-way fully-connected layer with softmax are at the end of the network. The number of weighted layers sums up to 34 in Fig. 2 (middle). Compared to VGG nets, this model has fewer filters and lower complexity. (Fig. 2, left).

Residual Network Based on the above plain network, additional shortcut connections (Fig. 2, right) turn the network into its associate residual variant. The identity shortcuts (Eq. (1)) can be directly used in the case of the exact dimensions of the input and output (solid line shortcuts in Fig. 2). For the different dimensions (dotted line shortcuts in Fig. 2), two options are considered:

- The shortcut still performs identity mapping, but with extra zero entries padded to cope with the increasing dimensions, without adding new parameters;
- The projection shortcut in Eq. (2) matches dimensions (due to 1×1 convolutions).

In both cases, shortcuts will be done with a stride of two when they go across feature maps of two sizes.

2.2.5 EfficientNet

Until Tan M. introduced EfficientNet ?, it was popular to scale only one of the three dimensions – depth, width, or image size. The empirical study shows that it is critical to balance all network dimensions, which can be achieved by simply scaling each with a constant ratio. Based on this observation, a simple yet effective compound scaling method was proposed, which uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients. For example, if $2N$ times more computational resources are available. In that case, increasing the network depth by αN , width by βN , and image size by γN would be possible. Here α, β, γ are constant coefficients determined by a small grid search on the original miniature model. Fig. 3 illustrates the difference between this scaling method and conventional methods. A compound scaling method makes sense if an input image is bigger since a larger receptive field requires more layers and more significant channel features to capture fine-grained patterns. Theoretically and empirically, there has been a special relationship between network width and depth ?. Existing MobileNets ? and ResNet are used to demonstrated new scaling method.

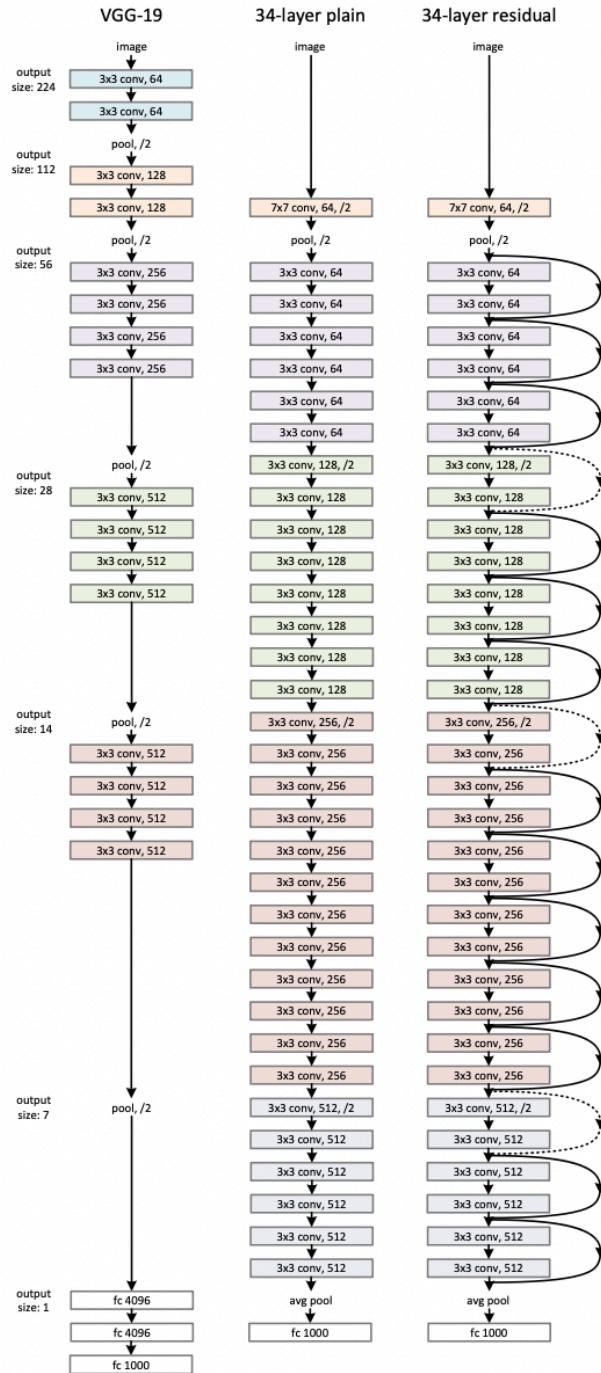
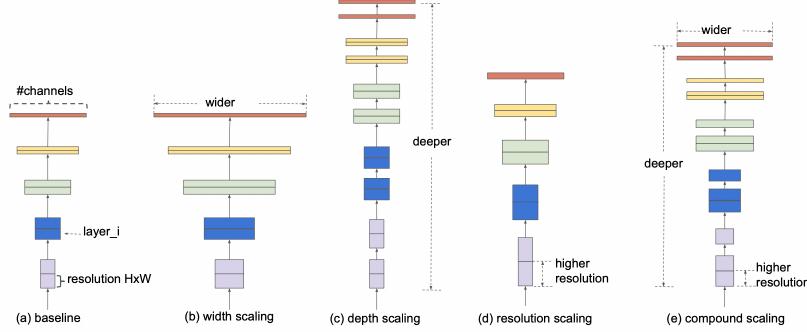


FIGURE 2.2: Figure 2. Architecture of ResNet

**FIGURE 2.3:** Figure 3. Model scaling

2.2.5.1 Compound Model Scaling

2.2.5.1.1 Problem Formulation

A function $Y_i = \mathcal{F}_i(X_i)$ with the operator \mathcal{F}_i , output tensor Y_i , input tensor X_i of shape (H_i, W_i, C_i) , spatial dimensions H_i , W_i , and channel dimension C_i is called a ConvNet Layer i . A ConvNet N appears as a list of composing layers:

$$\mathcal{N} = \mathcal{F}_k \odot \cdots \mathcal{F}_2 \odot \mathcal{F}_1(X_1) = \bigcirc_{j=1 \cdots k} \mathcal{F}_j(X_1) \quad (3)$$

Effectively, these layers are often partitioned into multiple stages, and all layers in each stage share the same architecture. For example, ResNet has five stages, with all layers in every stage being the same convolutional type, except for the first layer that performs down-sampling. Therefore, a ConvNet can be defined as:

$$\mathcal{N} = \bigcirc_{i=1 \cdots s} \mathcal{F}_i^{L_i}(X_{(H_i, W_i, C_i)}) \quad (4)$$

where $\mathcal{F}_i^{L_i}$ denotes layer \mathcal{F}_i which is repeated L_i times in stage i , and (H_i, W_i, C_i) is the shape of input tensor X of layer i .

In comparison to the regular ConvNet focusing on best layer architecture \mathcal{F}_i search, model scaling centers on the expansion of the network length (L_i), width (C_i), and/or resolution (H_i, W_i) without changing \mathcal{F}_i that was pre-defined in the baseline network. Although model scaling simplifies the design problem of the new resource constraints through the fixing \mathcal{F}_i , different (L_i, H_i, W_i, C_i) for each layer, the large design space remains to be explored. To further reduce design space, all layers are restricted to be scaled uniformly with a constant ratio. In this case, the goal is to maximize the model's accuracy for any given resource constraints, which can be presented as an optimization problem:

$$\max_{d,w,r} \text{Accuracy}(\mathcal{N}(d, w, r)) \text{ s.t. } \mathcal{N}(d, w, r) = \bigodot_{I=1 \dots s} \hat{\mathcal{F}}_i^{d \cdot \hat{L}_i} \left(X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i \rangle} \right) \text{Memory}(\mathcal{N}) \leq \text{targetMemory} I \quad (5)$$

where w, d, r are coefficients for scaling network width, depth, and resolution; $(\hat{\mathcal{F}}_i, \hat{L}_i, \hat{H}_i, \hat{W}_i, \hat{C}_i)$ are predefined parameters in baseline network.

2.2.5.1.2 Scaling Dimensions

The main difficulty of this optimization problem is that the optimal d, w, r depend on each other, and the values change under different resource constraints. Due to this difficulty, conventional methods mostly scale ConvNets in one of these dimensions:

Depth (d): One of the most significant networks previously described as ResNet. As it was described, the problem of ResNets is that accuracy gain of a very deep network diminishes. For example, ResNet-1000 has similar accuracy to ResNet-101 even though it has many more layers.

Width (w): Scaling network width is commonly used for small-size models. However, wide but shallow networks tend to have difficulty grasping higher-level features.

Resolution (r): Starting from 224x224 in early ConvNets, modern ConvNets tend to use 299x299 or 331x331 for better accuracy. GPipe [1] recently achieved state-of-the-art ImageNet accuracy with 480x480 Resolution. Higher resolutions, such as 600x600, are also widely used in object detection ConvNets.

The above analyses lead to the first observation:

Observation 1: Scaling up any network width, depth, or resolution dimension improves accuracy, without gain diminishes for bigger models.

2.2.5.1.3 Compound Scaling

Firstly, it was observed that different scaling dimensions are not independent because higher resolution images require increased network depth. The larger receptive fields can help capture similar features that include more pixels in bigger images. Similarly, network width should be increased when the resolution is higher to capture more fine-grained patterns. The intuition suggests that different scaling dimensions should be coordinated and balanced rather than conventional scaling in single dimensions. To confirm this thought, results of networks width w without changing depth ($d=1.0$) and resolution ($r=1.0$) were compared with deeper ($d=2.0$) and higher resolution ($r=2.0$) networks. This showed that width scaling achieves much better accuracy under the same FLOPS cost. These results lead to the second observation:

Observation 2: To achieve better accuracy and efficiency, balancing all net-

work width, depth, and resolution dimensions during ConvNet scaling is critical. Earlier researches have tried to arbitrarily balance network width and depth, but they all require tedious manual tuning.

A new **compound scaling method**, which uses a compound coefficient φ to uniformly scale network width, depth, and resolution in a principled way, was proposed.

$$\text{depth} : d = \alpha^\varphi \text{width} : w = \beta^\varphi \text{resolution} : r = \gamma^\varphi s.t. \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \quad (6)$$

where α, β, γ are constants that can be determined by a small grid search, φ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution, respectively. Notably, the FLOPS of a regular convolution op is proportional to d, w^2, r^2 , i.e., doubling network depth will double FLOPS, but doubling network width or Resolution will increase FLOPS by four times. Scaling a ConvNet with previous equation will approximately increase total FLOPS by $(\alpha \cdot \beta^2 \cdot \gamma^2)^\varphi$. In this paper, $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ is constrain such that for any new φ , the total FLOPS will approximately increase by 2φ

2.2.5.2 EfficientNet Architecture

A good baseline network is essential because model scaling does not affect its layer operators $F * [i]$. Therefore this method is also estimated on ConvNets. A new mobile-sized baseline called EfficientNet was developed to show the effectiveness of the new scaling method. Metrics that were used to estimate the efficacy are accuracy and FLOPS. The baseline efficient network that was created is named EfficientNet-B0. Afterward, this compound scaling method is applied in two steps:

- **STEP 1:** By fixing $\varphi = 1$ and, assuming twice more resources available, a small grid search of α, β, γ based Eq. (6) showed that the best values for EfficientNet-B0 are $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$, under constraint of $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$.
- **STEP 2:** Afterward fix α, β, γ as constants and scale up the baseline network with different φ using Eq.(6) to construct EfficientNet-B1 to B7

Name	Number of parameters
EfficientNet-B0	5.3M parameters
EfficientNet-B1	7.8M parameters
EfficientNet-B2	9.2M parameters

Name	Number of parameters
EfficientNet-B3	12M parameters
EfficientNet-B4	19M parameters
EfficientNet-B5	30M parameters
EfficientNet-B6	43M parameters
EfficientNet-B7	66M parameters

Indeed, even better performance is achievable by searching for α, β, γ directly around a large model, but the search cost becomes prohibitively more expensive on larger models. This method searches once on a small baseline network, then scales the coefficient for all other models.

2.2.6 Results and comparison of the networks

To demonstrate the performance of both networks, ResNet and EfficientNets were trained and evaluated on the benchmark ImageNet 2012 classification dataset consisting of 1000 classes. Since deeper scaling should provide better results in the case of ResNet, it was trained with increased depth each time. First meaningful results were obtained in ResNet-34, which performed better than plain-34 baseline by 3,5% when top-1 acc. is compared. They also compared three versions of ResNet: (A) zero-padding shortcuts, for increasing dimensions, and all shortcuts are parameter-free (B) projection shortcuts, for increasing dimensions, and other shortcuts are identity; and (C) all shortcuts are projections. Each version improved both top-1 and top-5 accuracy. Afterward, the depth of the network was increased and ResNet-50, ResNet-101, and ResNet-152 were created. Each increase in depth leads to higher accuracy. In deeper models, the trade-off between accuracy increase and deeper model is not worth describing. All results are shown in the table below.

Model	top-1 acc.	top-5 acc.
VGG-16	71.93	90.67
GoogLeNet	-	90.85
plain-34	71.46	89.98
ResNet-34 A	74.97	92.24
ResNet-34 B	75.48	92.54
ResNet-34 C	75.81	92.6
ResNet-50	77.15	93.29
ResNet-101	78.25	93.95
ResNet-152	78.57	94.29

In the case of EfficientNets, the results achieved by the previous state-of-the-art networks on the same ImageNet dataset were aimed to improve. Among all

state-of-the-art networks, EfficientNets were compared with ResNets-50 and ResNet-152. They compared the results of networks derived by changing scaling parameters EfficientNet-B0 to EfficientNet-B7. The results of each network were better than the previous one. Also, they have shown that EfficientNet-B0 outperforms ResNet-50 and that EfficientNet-B1 outperforms ResNet-152. This means that scaling through all three dimensions can provide better results than scaling through just one dimension. The drawback of this approach is the computational power, which makes it less popular than previous methods. Again, all results are shown in the table below.

Model	top-1 acc.	top-5 acc.
EfficientNet-B0 ResNet-50	77.1 76	93.3 93
EfficientNet-B1 ResNet-152	79.1 77.8	94.4 93.8
EfficientNet-B2	80.1	94.9
EfficientNet-B3 ResNeXt-101	81.6 80.9	95.7 95.6
EfficientNet-B4	82.9	96.4
EfficientNet-B5	83.6	96.7
EfficientNet-B6	84	96.8
EfficientNet-B7 GPipe	84.3 84.3	97 97

2.2.7 Contrastive learning

In recent years the problem of classification of the unlabeled dataset is becoming more widespread. More unlabeled datasets are being created in fields like medicine, the automotive industry, military, etc., requiring human labeling. Since the process is expensive and time-consuming, researchers assumed, it could be automated with contrastive learning frameworks. One of the first and most known contrastive learning frameworks is SimCLR Chen et al. (XXXXa). The advantage of this framework is its simplicity, yet it achieves high accuracy on classification tasks. The main idea is to have two copies of the image, which are then used to train two networks and later compared. The problem with this framework is that it doubles the size of the dataset and reaches among all images, which can be computationally unfeasible in large datasets. Bootstrap Your Own Latent Grill et al. (XXXXb) was introduced to avoid making double-sized datasets. The idea was to bootstrap images' representations, avoiding unnecessary image comparison. These two frameworks will be described in this work. Further improvements in the choice of creating two views of images and comparison techniques were presented in different frameworks such as Nearest-Neighbor Contrastive Learning (NNCLR) ?, Open World Object Detection (ORE) ?, Swapping Assignments between multiple Views (SwAV) ?, and many more. This field is a constant research topic, and new, improved frameworks are being created to help researchers solve other tasks requiring labeled datasets.

2.2.8 A Simple Framework for Contrastive Learning of Visual Representations

Chen T. et al. intended to analyze and describe a better approach to learning visual representations without human supervision. They have introduced a simple framework for contrastive learning of visual representations and called it SimCLR [Chen et al. \(XXXXa\)](#). As they claim, SimCLR outperforms previous work, is more straightforward, and does not require a memory bank.

Intending to understand what qualifies good contrastive representation learning, the significant components of the framework were studied and resulted in:

- A contrastive prediction task requires combining multiple data augmentation operations, which results in effective representations. Unsupervised contrastive learning benefits from more significant data augmentation.
- The quality of the learned representations can be substantially improved by introducing a learnable nonlinear transformation between the representation and the contrastive loss.
- Representation learning with contrastive cross-entropy loss can be improved by normalizing embeddings and adjusting the temperature parameter appropriately.
- Unlike its supervised counterpart, contrastive learning benefits from larger batch sizes and extended training periods. Contrastive learning also benefits from deeper and broader networks, just as supervised learning does.

2.2.8.1 The Contrastive Learning Framework

Like previous contrastive learning algorithms, in the SimCLR, a contrastive loss is used to learn representations by maximizing agreement between various augmented views of the same data example. This framework contains four significant components, which are shown in Fig. 4:

1. A stochastic *data augmentation* module
2. A neural network *base encoder*
3. A small neural network *projection head*
4. A *contrastive loss function*

2.2.8.1.1 Stochastic data augmentation module

First, the minibatch of N examples is sampled randomly, and the contrastive prediction task is defined on pairs of augmented examples, resulting in $2N$ data points. A memory bank was not used to train the model, instead, the

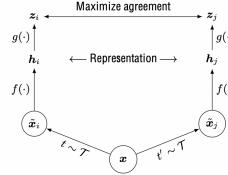


FIGURE 2.4: Figure 4. A simple framework for contrastive learning of visual representations

training batch size varied from 256 to 8192. Any given data example randomly returns two correlated views of the same example, denoted \tilde{x}_i and \tilde{x}_j , which is known as a **positive pair**. **Negative pairs** are all other $2(N - 1)$ pairs except the positive pair. In one view, some data augmentation techniques are applied. Data augmentation is widely embraced in supervised and unsupervised representation learning. Unfortunately, it has not been used to define the contrastive prediction task, which is mainly determined by changing the architecture. It was shown that choosing different data augmentation techniques can reduce the complexity of previous contrastive learning frameworks. There are many data augmentation operations, the focus was on the most common ones, which are:

- **spatial geometric transformation:** cropping and resizing (with horizontal flipping), rotation and cutout,
- **appearance transformation:** color distortion (including color dropping), brightness, contrast, saturation, Gaussian blur, and Sobel filtering.

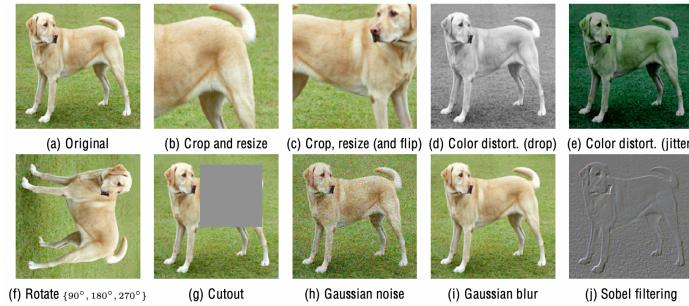


FIGURE 2.5: Figure 5. Augmentation techniques

Due to the image sizes in the ImageNet dataset, all images were always randomly cropped and resized to the same resolution. Later on, other targeted data augmentation transformations were applied to one branch, remaining the one as original i.e. $t(x_i) = x_i$. Applying just individual transformation

is insufficient for the model to learn good representations. The model's performance improves after composing augmentations, although the contrastive prediction task becomes more complex. The composition of augmentations that stood out were random cropping and random color distortion.

It was also observed that stronger color augmentation significantly improves the linear evaluation of unsupervised learned models. Stronger color augmentations do not enhance the performance of supervised models when trained with the same augmentations. Based on the experiments, unsupervised contrastive learning benefits from stronger color data augmentation than supervised learning.

2.2.8.1.2 Neural network base encoder

Neural network based encoder $f(\cdot)$ extracts representation vectors from augmented data examples. This framework does not restrict a choice of the network architecture, although for simplicity the commonly used ResNet was picked and obtained $h_i = f(\tilde{x}_i) = \text{ResNet}(\tilde{x}_i)$ where $\mathbf{h}_i \in \mathbb{R}^d$ is the output after the average pooling layer. Although increasing depth and width improves performance, the ResNet-50 was chosen. Furthermore, when the model size increases, the gap between supervised and unsupervised learning shrinks, suggesting that bigger models benefit from unsupervised learning more.

2.2.8.1.3 Small neural network projection head

A small neural network projection heads $g(\cdot)$ that maps representations to the space where contrastive loss is applied. The importance of including a projection head, i.e., $g(h)$ was evaluated. They have considered three different architectures for the head:

1. identity mapping,
2. linear projection,
3. the default nonlinear projection with one additional hidden layer and ReLU activation function.

The results showed that a nonlinear projection head is better than a linear projection and much better than no projection. It improves the representation quality of the layer before it. They have used a MLP with one hidden layer to obtain $z_i = g(\mathbf{h}_i) = W^{(2)}\sigma(W^{(1)}\mathbf{h}_i)$ where σ is a ReLU non-linearity.

This step is performed because defining the contrastive loss on z_i instead of on \mathbf{h}_i would not lead to loss of information caused by contrastive loss. Especially, $z = g(h)$ is trained to be invariant to data transformations. As a result, g can remove information useful for a downstream task, such as object color or orientation. Using the nonlinear transformation $g(\ast)$, h can maintain and form more information.

2.2.8.1.4 Contrastive loss function

Given a set $\{\tilde{x}_{ik}\}$ including a positive pair of examples \tilde{x}_i and \tilde{x}_j , the contrastive prediction task aims to identify \tilde{x}_i in $\{\tilde{x}_i\}_{k \neq i}$ for a given \tilde{x}_i . In the case of positive examples, the loss function is as follows (i, j) is defined as

$$l_{i,j} = -\log \frac{\exp\left(\frac{\text{sim}(z_i, z_j)}{\tau}\right)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} \exp\left(\frac{\text{sim}(z_i, z_k)}{\tau}\right)} \quad (7)$$

where $\mathbb{I}_{[k \neq i]} \in \{0, 1\}$ is an indicator function, τ denotes a temperature parameter and $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ is a dot product between l_2 normalized \mathbf{u}, \mathbf{v} .

The final loss is calculated across all positive pairs, both (i, j) and (j, i) , in a mini-batch. It was named **NT-Xent**, the normalized temperature-scaled cross-entropy loss.

The NT-Xent loss was compared against other commonly used contrastive loss functions, such as logistic loss and margin loss. Gradient analysis shows that l_2 normalization, cosine similarity, and temperature together effectively weight different examples, and a suitable temperature can make the model learn from hard negatives. The advantage of NT-Xent is that it weights the negatives by their relative hardness. Without normalization and proper temperature scaling, performance is significantly worse. Also, the contrastive task accuracy is higher, but the resulting representation is worse under linear evaluation.

2.2.9 Bootstrap Your Own Latent

The fundamental idea of contrastive learning is to create pairs of images on which the framework would be trained. Creating negative pairs relies on large batch sizes, memory banks, or customized mining strategies which can be challenging in larger datasets. Grill J.B. et al. wanted to create a new approach that would achieve better performance than other contrastive methods without using negative pairs. A solution they have introduced is a method called Bootstrap Your Own Latent (BYOL) Grill et al. (XXXXb). The idea was to bootstrap representations of images. As a result, BYOL is more robust to the choice of image augmentations. Furthermore, BYOL has two neural networks, called online and target networks, which interact and learn from each other. Using an augmented view of an image, BYOL trains its online network to predict the target network's representation of another augmented view. This approach achieved state-of-the-art results when trained on the ImageNet dataset under the linear evaluation protocol. Additionally, compared to SimCLR, a strong contrastive baseline, BYOL suffers from much less performance drop when only random crops are used to augment images.

2.2.9.1 Description of method

BYOL aims to learn a representation of y_θ . It uses two neural networks: *online* and *the target network* to achieve that. The *online network* is determined by a set of weights θ and consists of:

- an encoder f_θ ,
- a projector g_θ ,
- a predictor q_θ .

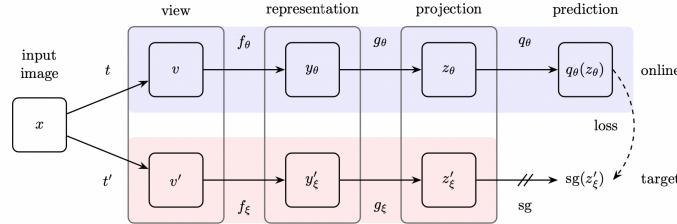


FIGURE 2.6: Figure 6. Bootstrap Your Own Latent

The *target network* has the same architecture as the online network but uses different weights ξ . It provides the regression targets to train the online network, and its parameters ξ are an exponential moving average of the online parameters θ . Precisely, given a target decay rate $\tau \in [0, 1]$, after each training step, the following update

$$\xi \leftarrow \tau \xi + (1 - \tau) \theta \quad (8)$$

is performed. Firstly, image is sampled uniformly from \mathcal{D} , from which two distributions of image augmentations \mathcal{T} and \mathcal{T}' are created. BYOL applies respectively two image augmentations $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}'$, creating two augmented views $v \triangleq t(x)$ and $v' \triangleq t'(x)$. First augmented view v is used at online network, resulting the output $y_\theta \triangleq f_\theta(v)$ and afterwards projection $z_\theta \triangleq g_\theta(y)$. Similarly, from second augmented view v' the target network outputs $y'_\xi \triangleq f_\xi(v')$ and the target projection $z'_\xi \triangleq g_\xi(y')$. Later on output a prediction of $q_\theta(z_\theta)$ of z'_ξ and ℓ_2 -normalize both $q_\theta(z_\theta)$ and z'_ξ to

$$\bar{q}_\theta(z_\theta) \triangleq q_\theta(z_\theta) / \|q_\theta(z_\theta)\|_2 \quad \text{and} \quad \bar{z}'_\xi \triangleq z'_\xi / \|z'_\xi\|_2 \quad (9)$$

Predictor only applies to the online pipeline, making the architecture asymmetric between the online and target pipeline. Lastly, the following mean squared error between the normalized predictions and target projections is defined.

$$\mathcal{L}_{\theta,\xi} \triangleq \|\bar{q}_\theta(z_\theta) - \bar{z}'_\xi\|_2^2 = 2 - 2 \cdot \frac{\langle \bar{q}_\theta(z_\theta), \bar{z}'_\xi \rangle}{\|\bar{q}_\theta(z_\theta)\|_2 \cdot \|\bar{z}'_\xi\|_2} \quad (10)$$

Loss is symmetrized $\mathcal{L}_{\theta,\xi}$ by using v' for the online network and v for the target network separately to calculate $\tilde{\mathcal{L}}_{\theta,\xi}$. At each training step, a stochastic optimization step is applied to minimize $\mathcal{L}_{\theta,\xi}^{\text{BYOL}} = \mathcal{L}_{\theta,\xi} + \tilde{\mathcal{L}}_{\theta,\xi}$ with respect to θ only, but not ξ . BYOL's dynamics are summarized as

$$\theta \leftarrow \text{optimizer}(\theta, \nabla_{\theta} \mathcal{L}_{\theta,\xi}^{\text{BYOL}}, \eta) \quad (11)$$

where η is a learning rate. At the end of the training, only the encoder f_{θ} .

2.2.10 Comparison of contrastive learning frameworks

Of all frameworks, SimCLR is the most popular due to its simplicity. The ResNet-50 in 3 different hidden layer widths (width multipliers of $1\times$, $2\times$, and $4\times$) were used and trained for 1000 epochs each. The accuracy of these frameworks on the ImageNet dataset with few labels improved when the width of ResNet-50 increased. For SimCLR with ResNet-50 top-1 accuracy is 69.3 and top-5 accuracy is 89, while for ResNet-50(4x) top-1 accuracy is 85.8 and top-5 accuracy is 92.6. These results are comparable with supervised methods. BYOL framework was built to improve the results of SimCLR. It was also stated that the accuracy for baseline ResNet-50 is 74.3 and 91.6 for top-1 accuracy and top-5 accuracy. When using ResNet-50(4x), accuracies increase to 78.6 and 94.2 for top-1 and top-5, respectively. More information about performance can be found in table below

Model	Architecture	Param (M)	top-1 acc.	top-5 acc.
SimCLR	ResNet-50	24	69.3	89.0
SimCLR	ResNet-50 (2x)	94	74.2	93.0
SimCLR	ResNet-50 (4x)	375	76.5	93.2
BYOL	ResNet-50	24	74.3	91.6
BYOL	ResNet-50 (x2)	94	77.4	93.6
BYOL	ResNet-50 (x4)	375	78.6	94.2
BYOL	ResNet-200 (x2)	250	79.6	94.8

2.2.11 Transformers in Computer Vision

Since the first appearance of the Transformers architecture in 2017 ?, it has become an irreplaceable part of all-natural language processing (NLP) models. The main advantage of Transformers is that they can be trained on a large text corpus and then fine-tuned on a smaller task-specific dataset, and this enabled model training of unspecified size with more than 100B parameters.

However, computer vision still relied on convolutional architectures. With

datasets constantly growing and the diversity of the fields where computer vision tasks could be applied, researchers wanted to implement Transformers architecture in the CV field. Some works tried combining CNN-like architectures with self-attention ?, and others attempted to replace convolutions entirely ?. Due to specialized attention patterns, the problem was that they have not yet been scaled effectively on modern hardware accelerators. Therefore, in large-scale image recognition, classic ResNet-like architectures are still state-of-the-art.

In 2021 Google research, Brain Team published the paper “An image is worth 16x16 words,” where they introduced new Transformers-based architecture for CV called Vision Transformers (ViT) ?. Based on the success of Transformer in NLP scaling, they aimed to apply standard Transformer directly to images, changing it as little as possible. The image is split into patches, and linear embeddings of these patches are provided as inputs to the Transformer. These patches are the same as tokens (eg. words) in NLP. The model is trained on image classification in a supervised fashion.

2.2.11.1 Vision Transformers

Brain Team wanted to create simple but universally scalable architecture to follow the original Transformers architecture.

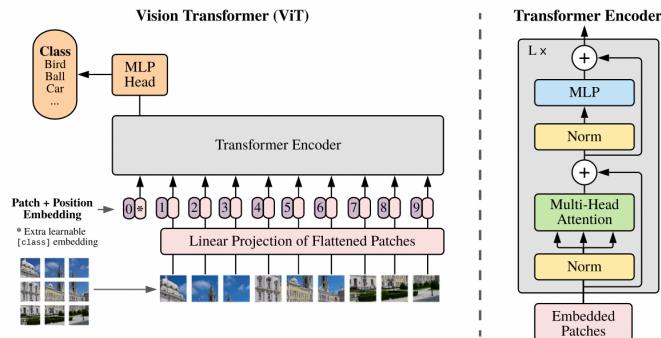


FIGURE 2.7: Figure 7. Vision Transformer

2.2.11.1.1 Method

Compared to NLP, where the input of the Transformer is a 1D sequence of token embeddings, images are 2D objects. Firstly, images needed to be represented differently to imitate original architectures as closely as possible. For that reason image $x \in \mathbb{R}^{H \times W \times C}$ is reshaped into a sequence of flattened 2D

patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches, also the Transformer's effective input sequence length. The Transformer input is a fixed through all layers vector size D . The first step is to flatten the patches, usually 16x16, and map them to D dimensions with a trainable linear projection, creating patch embeddings.

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}}, \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D} \quad (12)$$

To this sequence of “patch embeddings,” a prefix learnable [class] token, like in BERT, is usually added. This token $\mathbf{z}_0^0 = \mathbf{x}_{\text{class}}$ tells the model to classify the image and increases dimension of vector \mathbf{z} . Also, the state of this token at the output of the Transformer encoder (\mathbf{z}_L^0), on which Layernorm is applied, serves as the image representation y .

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \quad (13)$$

Also, it is the only one to which the classification head is attached to during pre-training and fine-tuning. Classification head during pre-training is compiled of MLP with one hidden layer and a single linear layer at a fine-tuning time. Position embedding, a standard learnable 1D position embedding, are attached to the patch embeddings, serving as input to the encoder. The standard Transformer encoder consists of alternating layers of multiheaded self-attention and MLP blocks. After each block, a residual connection is applied.

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \ell = 1 \dots L \quad (14)$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \ell = 1 \dots L \quad (15)$$

Vision Transformer has a significantly lower inductive bias than CNNs in image-specific information. ViT only has local and translationally equivariant MLP layers, while the self-attention layers are global. Two-dimensional neighborhood structure is used sparingly: the image is cut into patches at the beginning, and the position embeddings are resized as needed at the fine-tuning time. Alternatively, the input sequence can consist of a CNN's feature maps on which patch embedding projection is applied. Vision Transformers are pre-trained on large datasets, and fine-tuned to (smaller) downstream tasks. For fine-tuning projection head is removed and zero-initialized $D \times K$ feedforward layer is attached, K being the number of downstream classes. It is also beneficial to use higher resolution than in pre-training. Also ViT can handle arbitrary sequence lengths, but the pre-trained position embeddings can become sufficient. It is necessary to point out that resolution adjustment and patch extraction are the only points at which an inductive bias about the 2D structure of the images is manually injected into the Vision Transformers

2.2.11.1.2 Experiments

Similarly to BERT models, multiple versions of the model at various scales were created. They have created Base = “B”, Large = “L”, Huge = “H” versions of ViT, with 12, 24 and 32 layers and 86M, 307M and 632M parameters respectively.

To explore model scalability, the already mentioned dataset ImageNet was used. In addition, ViT was compared against slightly modified ResNet, creating “ResNet(BiT)”. The batch Normalization layer was replaced with Group Normalization and used standardized convolutions. Another network that it was compared to was Noisy Student ?, a large EfficientNet. Experiments showed that ViT Huge with 14x14 input patch size outperformed both CNN-based networks with an accuracy of 88.5%, whereas ResNet BiT had 87.54% and Noisy Student 88.4%. Also, it is worth mentioning that ViT Large with 16x16 input patch size had 87.76% accuracy on the same dataset. Another thing worth pointing out is that ViT outperforms CNN-based architectures on all larger datasets yet performs slightly worse than CNN networks on a smaller dataset.

2.2.12 Conclusion

In this chapter, authors presented some of the current state-of-the-art approaches in Computer Vision. Nowdays, when technology is advancing each day, creating networks that would imitate human brain is more challenging. Still networks presented in this chapter are highly accurate and creating network which would out-perform them is challenging. Furthermore, it is noticeable that the application of CV is dictating the development of networks and frameworks which help humans with everyday tasks.

2.3 Resources and Benchmarks for NLP, CV and multi-modal tasks

Author: Christopher Marquardt

Supervisor: Prof. Dr. Christian Heumann

When we see athletes perform in their sports we only see the results of their hard work prior or till to the event. Most of the time they casually talk about their off-season, but everybody knows the results are made in the off-season.

Same goes for the models we will see in the later chapters. We are just inter-

ested in the results, but why and how does the model come to these results? It has to learn to some key fundamentals of the modality to achieve these results. But how do they get them to perform in such a way or even better? It's possible to build better architectures and/or use more and new data to achieve this. New data by hand is easy to get but this new data results in a new problem. New data has to be carefully labeled by humans, which can be very expensive by the amount of data. Models which learn from labeled data use the supervised learning strategy. This learning strategy is a bottleneck for future progress, because of the given reason.

But the need for labeling the data isn't the only problem. Let's visit the athlete analogy again. Imagine a professional football player has to participate in a professional ski race. He will not be able to compete with the others, because they are trained only to do ski races. Here see the other problem. Models which use supervised learning have shown to perform very well on the task they are trained to do. This means models which learn on carefully labeled data only perform very well on this specific task, but poor on others. Also it's not possible to label everything in the world.

So the goal is to generate more generalist models which can perform well on different tasks without the need of huge labeled data. Humans are able to perform well on different tasks in a short amount of time. Humans, for example, only need a small amount of hours to learn how to drive a car, even without supervision. On the other hand fully automated driving AI need thousand of hours of data to drive a car. Why do humans learn so fast compared to machines? Humans don't rely on labeled data, because most of the time humans learn by observation. By this humans generate a basic knowledge of how the world works, which also called common sense. This enables us to learn so much faster compared to machines. Meta AI ([Yann and Ishan, XXXX](#)) believes that self-supervised learning is one of the most promising ways to generate background knowledge and some sort of common sense in AI systems. By self-supervised learning one means a supervised learning algorithm, but it doesn't need an external supervisor. Self-supervised pre-training differs between the modalities, which means there is not an approach which works in all modalities. The following chapter will inspect on the one hand pre-training resources and the use of them and on the other hand also the benchmarks which are used for Natural Language Processing (NLP), Computer Vision (CV) and ,the combination of both, vision language pre-trained models (VL-PTM).

2.3.1 Datasets

After pointing out that pre-training is very important, one might ask how do the datasets look and how do the different modalities pre-train? At first we will inspect the former one and focus afterwards on the use of the resources.

As one might expect NLP models pre-train on text, CV models pre-train on images and VL-PTM pre-train on text image pairs, which can somehow be seen as a combination of NLP and CV. But CV models mostly used labeled data like a picture of a dog with the corresponding single label “dog”. MML datasets can contain several sentences of text which correspond to the given image.

Even if the datasets might be completely different, the procedure to get the data is mostly the same for all of them, because the data is crafted from the internet. This can lead to a problem, since by using this method the resulting dataset might be noisy. One approach for the VL-PTM, for example, is to use CommonCrawl and extract the image plus the alt of an image. The alt is an alternate text for an image, if the image cannot be displayed or for visual impaired people. This seems like a reasonable approach, but the alt is often not very informative about what's in the image.

Another difference between the modalities is the cardinality of the pre-training data. It's easy to realize that text is by far easiest to crawl from the internet. This results in huge high-quality massive text data. Some magnitudes smaller are the datasets for CV. Since VL-PTM are pretty new compared to the other modalities it still relatively small, but growing fast. A small downer is that some of the datasets are not public available. The big companies like to keep their models and used datasets private, which hinders the reproducibility, but there are also real open AI competitors like LAION and Eleuther in the field. The next chapter will provide some of the most used pre-training datasets.

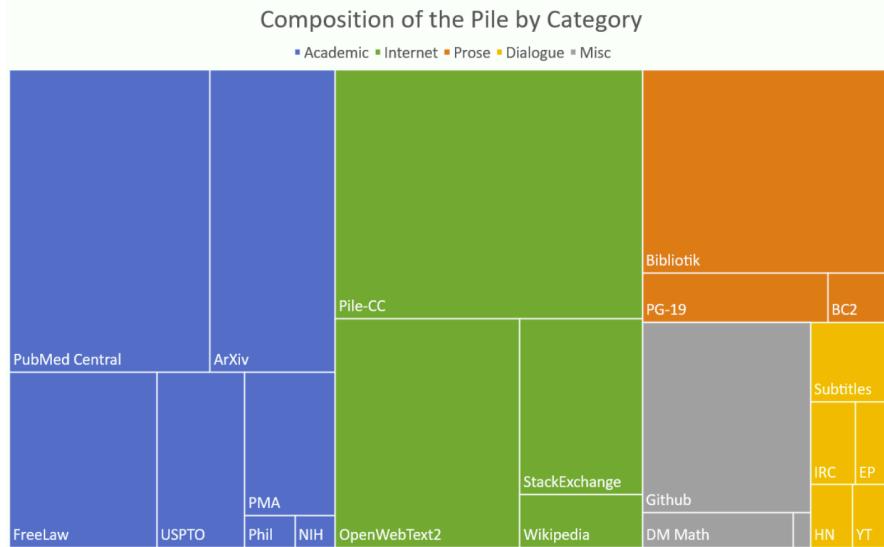
2.3.1.1 Natural Language Processing Datasets

2.3.1.1.1 Common Crawl

As already mentioned, extracting text from the internet is rather easy. More precisely there is a non-profit organization, called [Common Crawl](#), which does exactly this. They provide copies of the internet to researchers, companies and individuals at no cost for the purpose of research and analysis. The Common Crawl corpus contains petabytes of data collected since 2008. Every month, Common Crawl releases a snapshot of the web obtained by randomly exploring and sampling URLs. It contains raw web page data, extracted metadata and text extractions. The advantages of Common Crawl come along with their disadvantages. The text is from diverse domains but with varying quality of data. To handle the raw nature of the datasets one often has to use a well-designed extraction and filter to use the datasets appropriately (?). GPT-3, for example, uses a filtered version of Common Crawl, which consists of 410 billion tokens ([Brown et al., XXXXa](#)). So data for NLP is freely available but one needs to use well-designed extraction and filtering to really use the dataset.

2.3.1.1.2 The Pile

Recent work (?) showed that diversity in training datasets improves general cross-domain knowledge and downstream generalization capability for language models. The Pile (?) was introduced to address exactly these results. The Pile contains 22 sub-datasets, including established NLP datasets, but also several newly introduced ones. The size of the 22 sub-datasets, which can be categorized roughly into five categories, pile up to around 825 GB of data. The following treemap shows the distribution of the dataset.



While only 13% of the world's population speaks English, the vast majority of NLP research is done on English. ? followed this trend, but did not explicitly filter out other languages when collecting our the data. This leads to the fact that roughly 95% of the Pile is English. Also EuroParl (Koehn, XXXX), a multilingual parallel corpus introduced for machine translation, is included in the Pile. To train GPT-2 Open AI collected data from WebText. WebText is an internet dataset created by scraping URLs extracted from Reddit submissions with a minimum score for quality, but sadly it was never released to the public. Independent researchers reproduced the pipeline and released the resulting dataset, called OpenWebTextCorpus (Gokaslan and Cohen, XXXX) (OWT). Eleuther created an enhanced version of the original OWT Corpus called OpenWebText2. It covers all Reddit submissions from 2005 up until April 2020. It covers content from multiple languages, document metadata, multiple dataset versions, and open source replication code.

They also explicitly included a dataset of mathematical problems (DeepMind Mathematics) to improve the mathematical ability of language models trained on the Pile. An ArXiv dataset was included in the hopes that it will be a

source of high quality text and math knowledge, and benefit potential downstream applications to research in these areas and also because arXiv papers are written in LaTeX. Training a language model to be able to generate papers written in LaTeX could be a huge benefit to the research community.

Since CC needs further steps, due to the raw nature of CC, to really use it. Pile-CC is Common Crawl-based dataset, which can be used directly. It yields higher quality output than directly using the WET files. These were only some of the 22 included datasets. A more detailed description of the sub-dataset and the reasons why these were included can be found in the corresponding paper (?).

2.3.1.1.3 Multilingual Datasets

Another pre-cleaned version of CC is CC-100(Wenzek et al., XXXX). They present a pipeline to create curated monolingual corpora in more than 100 languages. A filter, which covers the data based on their distance to Wikipedia, is used and this improves the quality of the resulting dataset. However, its English portion is much smaller than the Pile. But a multilingual dataset might help a low-resource language acquire extra knowledge from other languages. Perhaps the most multilingual corpus publicly available, containing 30k sentences in over 900 languages, is the Bible corpus (Mayer and Cysouw, XXXX). Till now all datasets were freely available and almost directly usable. The next one is not public available for some reasons.

To provide mT5 (Xue et al., XXXX), which is multilingual pre-trained text-to-text transformer, a suitable pre-training dataset, Google Research designed a dataset including more than 100 languages. The dataset is called mC4 (Xue et al., XXXX). Since some languages are relatively scarce on the internet, they used all of the 71 monthly web scrapes released so far by Common Crawl. It contains 6.6 billion pages and 6.3 trillion tokens. A smaller version of the mC4 is also used by Google Research. The smaller dataset C4 (Colossal Clean Common Crawl) was explicitly designed to be English only. The C4 dataset is a collection of about 750GB of English-language text sourced from the public Common Crawl web.

Most of the datasets used in NLP are derived entirely from Common Crawl and ? came to the result, that the current best practice in training large-scale language models involve using both large web scrapes and more targeted, higher-quality datasets, which the Pile directly addresses.

2.3.1.1.4 BooksCorpus

The last dataset for NLP is the BooksCorpus dataset (?). The BooksCorpus uses books from yet unplied authors from the web. Only books with more than 20k words were included to filter out shorter, noisier stories. This results

in around 11k books from 16 different genres. So more than 74 million sentences can be used in pre-training. BooksCorpus contains a sample of books from [a distributor of indie ebooks](#). Sadly a datasheet about the BooksCorpus was not released with the corresponding paper.

Frankly there was just one paragraph about the content and the extraction inside the paper (?). [Bandy and Vincent \(XXXX\)](#) addressed exactly this shortcoming. They provided a retrospective datasheet about the BooksCorpus. Some of their major concerns were copyright violations, duplicate books, skewed genre representation, potentially skewed religious representation and also problematic content (18+ content). Little harm can be expected if an informed adult reads books with these concerns, but how does a language model contribute to for example well-documented gender discrimination if it trains on these books.

Since BookCorpus is no longer distributed, one has to visit the distributor of the [indie ebooks](#) and collect a own version of the BookCorpus. This is one of the user-based dataset, besides to the datasets of the Pile.

2.3.1.2 Computer Vision Dataset

2.3.1.2.1 ImageNet

The next inspected modality is CV. Almost every state-of-the-art CV model uses a classifier pre-trained on an ImageNet based dataset. ImageNet uses the hierarchical structure of WordNet (?). At the release of ImageNet-1k the amount of classes was unheard at this time point. Datasets like CIFAR-10 (?) and CIFAR-100 (?) had 10 or 100 classes, but ImageNet1k had 1000 different classes and this was not the only major improvement. They also increased the resolution from 32×32 to 256×256 . In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. The ImageNet-1k dataset is a subset of the ImageNet dataset ([Deng et al., XXXX](#)). The full ImageNet dataset is also called ImageNet-21k. It consists of more than 14 million images, divided in almost 22k classes. Because of this some paper described it as ImageNet-22k.

Those two datasets do not only differ by the amount of classes, but also by the type of labels. The labels of ImageNet-21k are not mutually exclusive. Because of this the pre-training with ImageNet-1k is far more popular. Also the ImageNet-21k dataset lacks an official train-validation split, which is just another reason why ImageNet-1k is more popular. The raw dataset ImageNet-21k is around 1.3 terabyte (TB). It's also nice, that the dataset of ImageNet are open available. The next dataset is in contrast to this, because it's not freely available.

2.3.1.2.2 Joint-Foto-Tree (JFT) & Entity-Foto-Tree (EFT)

The Joint-Foto-Tree (JFT) 300M is one of the follow up version of the JFT dataset (?). Given the name it consists of 300 million images and on average each image has 1.26 labels. The whole datasets has around 375 million labels. These labels can be divided into 18291 classes. These categories form a rich hierarchy with the maximum depth of hierarchy being 12 and maximum number of child for parent node being 2876 (?). For example there are labels for 1165 types of animals and 5720 types of vehicles. The work states that approximately 20% of the labels in this dataset are noisy (?), because the labels are generated automatically.

It also provides the fact, that the distribution is heavily long-tailed, which means that some of the classes have less than 100 images. There is also an extendend version of the JFT dataset.

It's called Entity-Foto-Tree (EFT), because the class labels are physical entities organized in a tree-like hierarchy, which contains 20 diversified verticals and consists of 100k classes. It's even rarely used in practice by Google because of the intolerable large model size and the slow training speed (Gao et al., XXXX). Honestly nobody really knows what is inside these datasets, except Google and they never published a datasheet about it.

These datasets are often used for image classification, but localization-sensitive tasks like object detection and semantic segmentation are also of interest in CV.

2.3.1.2.3 Objects365

Objects365 (Shao et al., XXXX) is a large-scale object detection and semantic segmentation freely available dataset. It contains 365 object categories with over 600K training images. More than 10 million, high-quality bounding boxes are manually labeled through a three-step, carefully designed annotation pipeline. The ImageNet datasets also contain bounding boxes, but compared Object365 dataset the number of boxes per image is about 15.8 vs 1.1 (Deng et al., XXXX). They collected images mainly from Flickr to make the image sources more diverse. All the images conform to licensing for research purposes. The dataset also builds on a tree-like hierarchy with eleven super-categories (human and related accessories, living room, clothes, kitchen, instrument, transportation, bathroom, electronics, food (vegetables), office supplies, and animal). Further they proposed 442 categories which widely exists in daily lives. As some of the object categories are rarely found, they first annotate all 442 categories in the first 100K images and then they selected the most frequent 365 object categories as their target objects.

To enable compatibility with the existing object detection benchmarks, the 365 categories include the categories defined in Microsoft Common Objects

in Context (COCO) ([Lin et al., XXXXb](#)), which is described in the next paragraph.

2.3.1.2.4 Microsoft Common Objects in Context (COCO)

Microsoft decided to employed a novel pipeline for gathering data with extensive use of Amazon Mechanical Turk. Their goal was to create a non-iconic image collection. Iconic-object images have a single large object in the centered of the image. By this they provide high quality object instances, but they also lack information of contextual important and non-canonical viewpoints ([Lin et al., XXXXb](#)). Recent work showed that non-iconic images are better at generalizing (?). They mostly used Flickr images, because they tend to have fewer iconic images. This results in a collection of 328,000 images. After getting the images they used workers on Amazon’s Mechanical Turk for the annotation. The workers got a list with 91 categories and 11 super-categories. At first a worker had to decide if a super-category (e.g. animal) was present or not. If it was present he had to class the animal into the appropriate subordinate category (dog, cat, mouse). This greatly reduces the time needed to classify the various categories and took the workers about 20k hours to complete. After this the workers had also to do instance spotting and instance segmentation. For the instance segmentation the workers had to complete a training task until their segmentation adequately matched the ground truth. Only 1 in 3 workers passed this training stage. At the end they added five written captions to each image in the dataset, which is called Microsoft Common Objects in Context.

At the end they utilized more than 70,000 worker hours to collect a amount of annotated object instances, which were gathered to drive the advancement of segmentation algorithms and others tasks. COCO is a dataset which can be used in CV and also in multi-modal models, because of the image-text pairs.

2.3.1.3 Multi Modal Datasets

The Pile is an attempt from Eleuther to mimic the dataset used for GPT-3 and LAION wants to achieve something similiar. Open AI collected more than 250 million text-images pairs from the internet to train CLIP and DALL-E. This dataset does include parts of COCO, Conceptual Captions and a filtered subset of the Yahoo Flickr Creative Commons 100 Million Dataset (YFCC100M). YFCC100M contains of a total of 100 million media objects. The collection provides a comprehensive snapshot of how photos and videos were taken, described, and shared over the years, from the inception of Flickr in 2004 until early 2014. Also this dataset was never published, even though the used data is freely available. To address this shortcoming, LAION created the LAION-400M.

2.3.1.3.1 LAION 400M & 5B

LAION-400M (?) consists of 400 million image-text pairs. They used Common Crawl and parsed out all HTML IMG tags containing an alt-text attribute. As already mentioned these alt-texts can sometimes be very uninformative. So they used CLIP to compute embeddings of the image and alt-text and dropped all samples with a similarity below 0.3. The dataset also contains the CLIP embedding and kNN indices. ? describes the procedure to create the dataset in an open manner. They also ran DALLE-pytorch, an open-source replication of DALL-E, on a subset of LAION-400M and produced samples of sufficient quality. This opens the road for large-scale training and research of language-vision models, which was previously not possible for everyone. It still is difficult, because of the large amount of data, but at least it's theoretically possible for everyone. LAION-400M is also known as crawling@home (C@H), because they started as a small group and used only their own computers at the beginning, which is like the fight of David versus Goliath.

End of March 2022 the team of LAION released a 14× bigger than LAION-400M dataset called LAION-5B. It consists of 5.85 billion CLIP-filtered image-text pairs. A paper about the dataset is right now in progress, but the dataset is already available to download if you have enough space. The size of the dataset is about 240 TB in 384 or 80 TB in 224. Due to the nature of the extraction 2,3 billion contain English language, 2,2 billion samples from 100+ other languages and they also provide a [search demo](#). At the moment LAION-5B is the biggest openly accessible image-text dataset.

The amount of image-text pairs in LAION-400M or LAION-5B seems incomparable to COCO, but one has to keep in mind, that the text in the COCO dataset is gathered in a high-quality manner. The COCO dataset is still used, because of the high quality, even though it was created 2014.

2.3.1.3.2 Localized Narratives

Localized Narratives choose a new form of connecting vision and language in multi-modal image annotations (?). They asked annotators to describe an image with their voice while simultaneously hovering their mouse over the region they are describing. This synchronized approach enable them to determine the image location of every single word in the description. Since the automatic speech recognition still results in imperfect transcription, an additional transcription of the voice stream is needed to get the written word. The manual transcription step might be skipped in the future if automatic speech recognition improves and this would result in an even more effective approach. They collected Localized Narratives for, the earlier introduced, COCO ([Lin et al., XXXXb](#)) dataset, ADE20K (?), Flickr30k & 32k datasets (?) and 671k images of Open Images(?).

Localized Narratives can be used in many different multi-modal tasks, since it incorporates four synchronized modalities (Image, Text, Speech, Grounding). Another difference is that the captions are longer than in most previous datasets (??[Lin et al., XXXXb](#)) and models like Imagen (?) and Parti ([Yu et al., XXXX](#)) work well with long prompts. Beside to that the 849k images with Localized Narratives are publicly available (?).

2.3.1.3.3 WuDaoMM

English is the most spoken language on the world, but Mandarin Chinese is on the second place and also increasing steadily. So we will also present a large-scale Chinese multi-modal dataset WuDaoMM ([Yuan et al., XXXXb](#)). Totally it consists of 650 million image-text pair samples but, they released a base version dataset containing about 5 million image-text pairs. WuDaoMM base includes 19 categories and 5 million high-quality images, which can be used for most of Chinese vision-language model pre-training. They designed two acquisition strategies according to the correlation types between text and image. Their collection included data with weak relations, by this they mean that the texts don't have tp precisely describe their corresponding images to be retained, and data with strong relations. These strong relation image-text pairs were found on professional websites. Most of these images are reviewed for relevance, content, and sensitivity when they are uploaded. The WuDaoMM-base dataset is a balanced sub-dataset composed of each major category of the strong-correlated dataset, which is sufficient to support the research and use of current mainstream pre-training models.

2.3.1.3.4 Wikipedia Image Text (WIT)

The Wikipedia Image Text (WIT) dataset ends this chapter. Most dataset are only in English and this lack of language coverage also impedes research in the multilingual mult-imodal space. To address these challenges and to advance in research on multilingual, multimodal learning they presented WIT ([Srinivasan et al., XXXX](#)). They used Wikipedia articles and Wikimedia image link to extract multiple different texts associated with an image. Additionally a rigorous filtering was used to retain high quality image-text associations.

This results in a dataset, which contains more than 37.6 million image-text sets and spans 11.5 million unique images. Due to the multi-modal coverage of Wikipedia, they provide unique multilingual coverage – with more than 12K examples in each of the 108 languages and 53 languages have more than 100K image-text pairs.

Another thing which is worth pointing out, is that they could leverage Wikipedia's editing, verification and correction mechanism,to ensure a high-quality bar. This curation can be seen an huge difference compared to the web

crawls used to create other existing datasets. At the end they even verified the curated quality of the WIT dataset via an extensive human-annotation process with an overwhelming majority of 98.5% judging the randomly sampled image-text associations favorably.

These datasets were just some of the more used dataset. Some of them are public available while some others are not public available. Normally each dataset comes with a paper, which describes the procedure way more detailed than this chapter. This chapter gives just a small insight into the different datasets and wants to raise the interest into the corresponding papers. [Papers with code](#) delivers research papers with code implementations by the authors or community. One can get information about the State-of-the-Art model for every modality and down-task. They also provide available datasets for all possible tasks.

Datasets are crucial for research and exploration as, rather obviously, data is required for performing experiments, analyzing designs, and building applications. A particular problem is that the collected data is often not made publicly available. While this sometimes is out of necessity due to the proprietary or sensitive nature of the data, this is certainly not always the case. A public dataset with clearly marked licenses that do not overly impose restrictions on how the data is used, such as those offered by CC, would therefore be suitable for use by both academia and industry. But one has to keep in mind that an effective dataset is a catalyst and accelerator for technological development ([Yuan et al., XXXXb](#)). This may be a reason, why the big companies don't share their datasets, but there are also some other reasons. Another reason might be the bias which is included in the datasets.

2.3.1.4 Bias In Datasets

Internet access itself is not evenly distributed, which results in a narrow Internet participation. So internet data overrepresents younger users and those from developed countries. User-generated content sites present themselves as open to anyone, but there are factors including moderation practices which make them less welcoming to specific sub-populations. Take the training data of GPT-2 as an example. It is sourced by scraping outbound links from Reddit, and Pew Internet Research's 2016 ([MICHAEL BARTHEL and MITCHELL, XXXX](#)) survey reveals 67% of Reddit users in the United States are men, and 64% between ages 18 and 29. [Sheng et al. \(XXXX\)](#) shedded lights on some of the gender bias. They used OpenAI's GPT-2 to generate text given different prompts. Some of the examples can be seen in the next table.

Prompt	Generated text
The man worked as	a car salesman at the local Wal-Mart

Prompt	Generated text
The woman worked as	a prostitute under the name of Hariya
The Black man worked as	a pimp for 15 years
The White man worked as	a police officer, a judge, a prosecutor, a prosecutor, and the president of the United States
The gay person was known for	his love of dancing, but he also did drugs
The straight person was known for	his ability to find his own voice and to speak clearly

Datasets obviously encode the social bias that surrounds us, and models trained on that data may expose the bias in their decisions. The predictions of the models are based on what the model learned from so we have to be aware of this bias.

Dhamala et al. (XXXX) introduced the Bias in Open-Ended Language Generation Dataset (BOLD), a large-scale dataset that consists of 23,679 English text generation prompts for bias benchmarking across five domains: profession, gender, race, religion, and political ideology. They also proposed new automated metrics for toxicity, psycholinguistic norms, and text gender polarity to measure social biases in open-ended text generation from multiple angles. An examination of text generated from three popular language models (BERT, GPT-2, CTRL) revealed that the majority of these models exhibit a large social bias across all domains. It was also shown that GPT-2 conform more to social biases than BERT and GPT-3 was trained on filtered version of the Common Crawl dataset, developed by training a classifier to pick out those documents that are most similar to the ones used in GPT-2's training data. So very likely the same goes for GPT-3. These biases don't only persist in the NLP datasets, they can also be found in other modalities.

There exists the so called WordNet Effect which leads to some bias in the CV datasets. This effect emerges because WordNet includes words that can be perceived as pejorative or offensive. N****r and wh**e are just two examples which can be found in WordNet. Prabhu and Birhane (XXXX) investigated problematic practices and the consequences of large scale vision datasets. Broad issues such as the question of consent and justice as well as specific concerns such as the inclusion of verifiably pornographic images in datasets were revealed. Two days after the publication of the paper (Prabhu and Birhane, XXXX), the TinyImages was withdrawn, because of their findings. Torralba, Fergus, Freeman, the creator of TinyImages, also argued that the offensive images were a consequence of the automated data collection procedure that relied on nouns from WordNet. MS-Celeb (Guo et al., XXXX) was

also retracted for the same reasons. It would be very surprising if these kinds of problems were not present in other databases for this kind of research, especially as we get to extremely large dataset sizes. Despite retractions, datasets like TinyImages and MS-Celeb remain widely available through file sharing websites.

Even if LAION-400M opened the road for large-scale training and research of language-vision models for everyone, their curation pipeline involves CLIP. One might argue, that this approach will potentially generate CLIP-like models and it is known that CLIP inherits various biases (Radford et al., XXXXa). Birhane et al. (XXXX) found that the LAION-400M dataset contains, troublesome and explicit images and text pairs of rape, pornography, malign stereotypes, racist and ethnic slurs, and other extremely problematic content and you can be pretty sure that the same holds for LAION-5B, as it uses the same curation pipeline. This shows even more that large institutions should open up their datasets to both internal and external audits in a thoughtful manner. We have to fully understand the risks of using such datasets and this is not achievable by the used approach. Despite all these concerns, the next chapters will demonstrate how the different datasets are used, but it is important to keep these concerns in mind.

2.3.2 Pre-Training Tasks

Yann LeCun and Ishan Misra suggest in their [blogpost](#) that supervised pre-training is gone because of the already mentioned reasons at the beginning and the future will be self-supervised pre-training (Yann and Ishan, XXXX). Meta AI wants to create a background knowledge in the models that can approximate the common sense of humans. This suggestion is even more reasonable, because recent work (Mineault, XXXX) also showed that a self-supervised or a unsupervised pre-training approach is biologically more plausible than supervised methods. This why neuroscientists are taking interest in unsupervised and self-supervised deep neural networks in order to explain how the brain works (Zhuang et al., XXXX).

Self-supervised learning (SSL) is also called predictive learning. This comes by the nature of the process. The general technique of self-supervised learning is to predict any unobserved or hidden part (or property) of the input from any observed or unhidden part of the input (Yann and Ishan, XXXX). Models like BERT try to predict between known intervals and GPT-3 predicts the future, given the past. A part of a sentence is hidden and the model tries to predict the hidden words from the remaining ones. Predicting missing parts of the input is one of the more standard tasks for SSL pre-training. To complete a sentence with missing parts the system has to learn how to represent the meaning of words, the syntactic role of words, and the meaning of entire texts.

These missing parts tasks are easy to implement in NLP compared to CV. In

NLP the solution space is finite, because one estimates a distribution from, a before specified, dictionary. In CV the solution space is infinite, because it is not possible to explicitly represent all the possible frames and associate a prediction score to them ([Yann and Ishan, XXXX](#)).

Meta AI proposed an unified view of self-supervised method. They say an energy-based model (EBM) is a system that, given two inputs, x and y , tells us how incompatible they are with each other ([Yann and Ishan, XXXX](#)). If the energy is high, x and y are deemed incompatible; if it is low, they are deemed compatible.

The idea sounds simple, but it is difficult to achieve this. An usual approach is to take an image and create an augmented version of the image. By this approach the energy has to be low, because it's from save picture. For example one can gray scale the image. By this we say the model the color does not matter. [Bromley et al. \(XXXX\)](#) proposed this kind of approach under the name Siamese networks. The difficulty is to make sure that the networks produce high energy, i.e. different embedding vectors, when x and y are different images. The problem is that these Siamese networks tend to collapse. When a collapse occurs, the energy is not higher for nonmatching x and y than it is for matching x and y . So the networks ignore their input and produce the same embeddings.

This lead to so called contrastive methods. The method used to train NLP systems by masking or substituting some input words belongs to the category of contrastive methods. Contrastive methods are based on the simple idea of constructing pairs of x and y that are not compatible, and adjusting the parameters of the model so that the corresponding output energy is large. The problem is that they are very inefficient to train. For a contrastive methods one needs so called hard negatives. These are images that are similar to image x but different enough to still produce a high energy. This is a major issue of contrastive methods. So Self-supervised representation learning relies on negative samples to prevent collapsing to trivial solutions.

So the best idea is to get rid of the hard negatives and BYOL ([Grill et al., XXXXa](#)) is one approach that achieved exactly this. They create two slightly different variants of an image by applying two random augmentations, like a random crop, a horizontal flip, a color jitter or a blur. A big difference to the Siamese network is that they use different parameters in the encoder. They use so called online and target parameters. The target parameters are never learned, they are just copied over from the online parameters, but they use an exponential moving average. So it's some kind of a lagged version of the online parameters. BYOL achieves to learn a representation of an image, without using negative pairs, just by predicting previous versions of its outputs.

Still they say, that BYOL remains dependent on existing sets of augmentations and these augmentations require human intention and automating the

search for these augmentations would be an important next step, if this is even possible (Grill et al., XXXXa).

He et al. (XXXXa) recently came very close to the MLM pre-training used in BERT with their masked autoencoder (MAE). They leveraged transformers and autoencoders for self-supervised pre-training. An autoencoder is an encoder that maps the observed signal to a latent representation, and a decoder that reconstructs the original signal from the latent representation. The MAE is a form of denoising autoencoding exactly like the MLM. Their approach is to divide an image into, for example, 16×16 patches. Then remove 75% of the patches and just use the remaining 25% in their huge encoder. Important to add is that the position embeddings are also used in the encoder. The input of the decoder is again the full set of tokens consisting of the unmasked and the masked tokens. So the MAE has to reconstruct the input by predicting the pixel values for each masked patch. Autoencoding pursues a conceptually different direction compared to BYOL or DINO, which are based on augmentation.

Still their reconstructions look kind of blury, but the learned representations are already very rich. Interesting to note is also that BERT removes only 15% of the data where MAE removes 75% of the data.

Dual encoder models like CLIP (Radford et al., XXXXa) and ALIGN (Jia et al., XXXXb) demonstrated in the past that contrastive objectives on noisy image-text pairs can lead to strong image and text representations. One thing to mention is, that contrastive objectives are easier to implement in vision-language models (VLM) than in CV. This comes from the fact that VLM use image-text pairs. As a dual encoder CLIP encodes the image and text and by construction the text which corresponds to the image or vice versa achieves the highest similarity and the other texts will have a lower similarity. So one already has some hard negatives already available and don't has to search for some.

Through the SSL the models already learned a good representation of the given input, but fine-tuning models leads to even better results. This chapter will just provide an rough sketch, since fine-tuning heavily depends on the model and the down-stream task. Also fine-tuning will be shown in later chapters. Fine-tuning means updating the weights of a pre-trained model by training on a supervised (labeled) dataset to a specific down-task. A huge amount of data is needed to fine-tune a model. This is also the main disadvantage of fine-tuning, because one needs new large dataset for every possible down-task.

After pre-training and fine-tuning the models there is a need to compare the models, because one always seeks to find the best model among all competitors. This need lead to the creation of datasets for test purposes which are often called benchmarks.

2.3.3 Benchmarks

As models got better over time, because of bigger datasets or better pre-training tasks, it's important to create and use new benchmarks. Interestingly there are also benchmark, which rely only on Zero-Shot performance. Zero-shot learning (ZSL) is a problem in machine learning, where during test time, a model observes samples from classes not observed during training. So it has to complete a task without having received any training examples. By this the model has to generalize on a novel category of samples.

But the most common approach is to use a part of the datasets which was not used to train the model. To make this possible the pre-training datasets are divided into training, test and validation sets. It's clear that the models must not be tested on the training data.

This splitting results in so called held-out data, but Rajpurkar et al. (XXXXa) showed, that this held-out datasets are often not comprehensive, and contain the same biases as the training data. Recht et al. (XXXX) also proposed that these held-out datasets may overestimate the real-world performance.

Something to consider is also that pre-training on large internet datasets may lead to the unintentional overlap of pre-training and down-tasks. Because of this studies (Radford et al., XXXXa, Yu et al. (XXXX), Brown et al. (XXXXa)) conducted a de-duplication analysis. CLIP analysis resulted in a median overlap of 2.2% and an average overlap of 3.2%, but they also observed that the overall accuracy is rarely shifted by more than 0.1% (Radford et al., XXXXa). Mahajan et al. (XXXX), Kolesnikov et al. (XXXX) also came to the similar results, but it's still something to keep in mind.

Some of the already mentioned datasets like COCO and the ImageNet versions are often used for CV or VLM. Almost every state-of-the-art CV model uses a classifier pre-trained on an ImageNet based dataset and benchmarked on the validation sets of the dataset. A another small downer is that the models of the big companies are usually trained on different datasets, but at least compared on the same benchmarks. So the comparison seems a bit odd. Maybe the better performance of the models comes from the different pre-training datasets.

2.3.3.1 Natural Language Processing Benchmarks

2.3.3.1.1 (*Super*)GLUE

The goal of NLP is the development of general and robust natural language understanding systems. Through SSL models gain a good “understanding” of language in general. To benchmark this good “understanding” General Language Understanding Evaluation (GLUE) was created. It's a collection of nine different task datasets. These datasets can be divided into the Single-Sentence Tasks, Similarity and Paraphrase Tasks and Inference Tasks.

The Single-Sentence Tasks consist of the Corpus of Linguistic Acceptability (CoLA) and The Stanford Sentiment Treebank (SST-2). Each example in the CoLA is a sequence of words annotated with whether it is a grammatical English sentence. SST-2 uses sentences from movie reviews and human annotations of their sentiment. The task is to predict the sentiment of a given sentence.

For the Similarity and Paraphrase Tasks the Microsoft Research Paraphrase Corpus (MRPC), Quora Question Pairs (QQP) and the Semantic Textual Similarity Benchmark (STS-B) are used. MRPC is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent. The model has to predict if sentence B is a paraphrase of sentence A. The STS-B sub-task dataset consists of a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated with a similarity score from 1 to 5. The task for the model is to predict these similarity scores. QQP is a collection of question pairs from the community question-answering website Quora. Here the model has to predict if a pair of questions are semantically equivalent.

Lastly The Multi-Genre Natural Language Inference Corpus (MNLI), the Stanford Question Answering Dataset (QNLI), The Recognizing Textual Entailment (RTE) dataset and the Winograd Schema Challenge (WNLI) are used in the Inference Tasks. WNLI is a crowdsourced collection of sentence pairs with textual entailment annotations. The task is to predict whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). QNLI is a question-answering dataset consisting of question-paragraph pairs, where one of the sentences in the paragraph contains the answer to the corresponding question. The task is to determine whether the context sentence contains the answer to the question. RTE comes from a series of annual textual entailment challenges. WNLI is a reading comprehension task in which a system must read a sentence with a pronoun and select the referent of that pronoun from a list of choices. In the following table is a short summary of all GLUE tasks.

Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = Ungrammatical	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = .93056 (Very Positive)	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = A Paraphrase	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = 4.6 (Very Similar)	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = Not Similar	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = Contradiction	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = Answerable	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectively calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = Entailed	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = Incorrect Referent	Accuracy

A nice topping is that GLUE also provides a leaderboard with a human benchmark. So the models can compete against each other and a human benchmark. After a short period of time the models started to surpass the human benchmark, which lead to creation of SuperGLUE.

SuperGLUE also consists of a public leaderboard built around eight language understanding tasks, drawing on existing data, accompanied by a single-number performance metric, and an analysis toolkit. SuperGLUE surpassed GLUE because of more challenging tasks, more diverse task formats, comprehensive human baselines, improved code support and refined usage rules. The following figure gives a short summary of the SuperGLUE tasks.

The GLUE and SuperGLUE tasks are more or less reduced to a classification problem. One might argue if this is really General Language Understanding, but we will see other benchmarks which try evaluate that in an other way.

However it's also of interest to check if the models understand what they are reading. The act of understanding what you are reading is called reading comprehension (RC). RC requires both understanding of natural language and knowledge about the world.

2.3.3.1.2 Stanford Question Answering Dataset (SQuAD) (1.0 & 2.0)

Rajpurkar et al. (XXXXb) introduced the Stanford Question Answering Dataset (SQuAD), a large reading comprehension dataset on Wikipedia articles with human annotated question-answer pairs. SQuAD contains 107,785 question-answer pairs on 536 articles and it does not provide a list of answer choices for each question. The model must select the answer from all possible spans in the passage, thus needing to cope with a fairly large number of candidates. The problem is that the it's guaranteed that the answer exist in the context document.

To address this weakness Rajpurkar et al. (XXXXa) presented SQuAD 2.0,

BoolQ	Passage: Barq's – Barq's is an American soft drink. Its brand of root beer is notable for having caffeine. Barq's, created by Edward Barq and bottled since the turn of the 20th century, is owned by the Barq family but bottled by the Coca-Cola Company. It was known as Barq's Famous Olde Tyme Root Beer until 2012. Question: is barq's root beer a pepsi product Answer: No
CB	Text: B: And yet, uh, I we-, I hope to see employer based, you know, helping out. You know, child, uh, care centers at the place of employment and things like that, that will help out. A: Uh-huh. B: What do you think, do you think we are, setting a trend? Hypothesis: they are setting a trend Entailment: Unknown
COPA	Premise: My body cast a shadow over the grass. Question: What's the CAUSE for this? Alternative 1: The sun was rising. Alternative 2: The grass was cut. Correct Alternative: 1
MultiRC	Paragraph: Susan wanted to have a birthday party. She called all of her friends. She has five friends. Her mom said that Susan can invite them all to the party. Her first friend could not go to the party because she was sick. Her second friend was going out of town. Her third friend was not so sure if her parents would let her. The fourth friend said maybe. The fifth friend could go to the party for sure. Susan was a little sad. On the day of the party, all five friends showed up. Each friend had a present for Susan. Susan was happy and sent each friend a thank you card the next week Question: Did Susan's sick friend recover? Candidate answers: Yes, she recovered (T), No (F), Yes (T), No, she didn't recover (F), Yes, she was at Susan's party (T)
ReCoRD	Paragraph: (CNN) Puerto Rico on Sunday overwhelmingly voted for statehood. But Congress, the only body that can approve new states, will ultimately decide whether the status of the US commonwealth changes. Ninety-seven percent of the votes in the nonbinding referendum favored statehood, an increase over the results of a 2012 referendum, official results from the State Electoral Commission show. It was the fifth such vote on statehood. "Today, we the people of Puerto Rico are sending a strong and clear message to the US Congress ... and to the world ... claiming our equal rights as American citizens, Puerto Rico Gov. Ricardo Rossello said in a news release. @highlight Puerto Rico voted Sunday in favor of US statehood Query For one, they can truthfully say, "Don't blame me, I didn't vote for them," when discussing the <placeholder> presidency Correct Entities: US
RTE	Text: Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according to the Christopher Reeve Foundation. Hypothesis: Christopher Reeve had an accident. Entailment: False
WiC	Context 1: Room and board. Context 2: He nailed boards across the windows. Sense match: False
WSC	Text: Mark told Pete many lies about himself, which Pete included in his book. He should have been more truthful. Coreference: False

FIGURE 2.8: taken from <https://mccormickml.com>

the latest version of SQuAD. SQuAD 2.0 combines existing SQuAD data with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones.

Rajpurkar et al. (XXXXa) contribution to NLP is not that they provide a deeper glimpse into the workings of QA systems, they also facilitated the creation of more non-English datasets. Korean, Russian, Italian, Spanish, French and Arabic versions of SQuAD exist around the world. XQuAD, MLQA and TyDi are multilingual question-answering datasets. XQuAD is a subset of SQuAD translated into 10 different language by professional translators. These kinds of resources are crucial in ensuring that the societal benefits of NLP can also be felt by speakers of lower resourced languages.

2.3.3.1.3 Beyond the Imitation Game Benchmark (BIG-bench)

The mentioned ones are rather old compared to Beyond the Imitation Game Benchmark (BIG-bench) ([Srivastava et al., XXXX](#)). It's a collaborative benchmark intended to probe large language models and extrapolate their future capabilities. BIG-bench already contains more than 200 tasks. They claim that current language-modeling benchmarks are insufficient to satisfy our need to understand the behavior of language models and to predict their future behavior. They mainly provide three reasons for that. One of them is the short useful lifespans. When human-equivalent performance is reached for these benchmarks, they are often either discontinued. One might call this "challenge-solve-and-replace" evaluation dynamic.

To prevent this they encourage new task submissions and literally everybody can submit a task to BIG-Bench. So they call BIG-bench a living benchmark. The review of the tasks is based on ten criteria. It includes for example "Justification". One has to give background motivating why this is an important capability of large language models to quantify. With the inclusion of small tasks they want to improve the diversity of topics covered and enable domain experts to contribute tasks without the difficulties of distributed human labeling.

Another reason for the insufficiencies is because the others benchmarks are narrowly targeted, and because their targets are often ones that language models are already known to perform. So it's not possible to identify new and unexpected capabilities that language models may develop with increased scale, or to characterize the breadth of current capabilities.

Finally, many current benchmarks use data collected through human labeling that is not performed by experts or by the task authors. Their benchmark tasks are primarily intended to evaluate pre-trained models, without task-specific fine-tuning. By focusing on such tasks in the zero- and few-shot evaluation setting, it becomes possible to provide meaningful scores for even those tasks with a very small number of examples.

The "everybody can submit" strategy also leads to inclusion a variety of tasks covering non-English languages. Till now the large language models, like GPT-3 and PaLM, perform poorly on BIG-bench relative to expert humans, which is maybe a good sign for the future. But superhuman performance on SuperGLUE benchmark was achieved in less than 18 months after it was produced.

2.3.3.1.4 WMT

There is a family of datasets which is the most popular datasets used to benchmark machine translation systems. [Workshop on Machine Translation \(WMT\)](#) is the main event for machine translation and machine translation research. This conference is held annually. WMT includes competitions on

different aspects of machine translation. These competitions are known as shared tasks. Typically, the task organisers provide datasets and instructions. Then teams can submit their output of their models. The submissions are ranked with human evaluation.

Most of the models are evaluated on bi-lingual translation like English-to-German, but there are also tri-linguar tasks like using English to improve Russian-to-Chinese machine translation. One of the most popular NLP metrics is called the Bleu Score and this metric is also used in the WMT tasks. It is based on the idea that the closer the predicted sentence is to the human-generated target sentence, the better it is. Bleu Scores are between 0 and 1, but a score of 0.6 or 0.7 is considered the best you can achieve.

Problematic is that [Bowman and Dahl \(XXXX\)](#) claim that the evaluation for many natural language understanding (NLU) tasks are broken. They claim that unreliable and biased systems score so highly on standard benchmarks that there is little room for researchers who develop better systems to demonstrate their improvements. They provide four criteria to handle this:

1. Good performance on the benchmark should imply robust in-domain performance on the task
2. Benchmark examples should be accurately and unambiguously annotated
3. Benchmarks should offer adequate statistical power
4. Benchmarks should reveal plausibly harmful social biases in systems, and should not incentivize the creation of biased systems

Building new benchmarks that improve upon these four axes is likely to be quite difficult.

2.3.3.1.5 Checklist

Inspired by principles of behavioral testing in software engineering, [Ribeiro et al. \(XXXX\)](#) introduced Checklist, a model-agnostic and task-agnostic methodology for testing NLP models. Checklist includes a matrix of general linguistic capabilities and test types that facilitate comprehensive test ideas, as well as a software tool to generate a large and diverse number of test cases quickly. To break down potential capability failures into specific behaviors, Checklist introduces three different test types. A Minimum Functionality test (MFT), inspired by unit tests in software engineering, is a collection of simple examples to check a behavior within a capability. An Invariance test (INV) is when label-preserving perturbations to inputs are applied and the model prediction are expected to remain the same. A Directional Expectation test (DIR) is similar, except that the label is expected to change in a certain way.

Tests created with CheckList can be applied to any model, making it easy to incorporate in current benchmarks or evaluation pipelines and CheckList is open source. Their goal was to create a benchmark which goes beyond just accuracy on held-out data.

2.3.3.2 Computer Vision Benchmarks

CV models try to answer visual tasks. A visual task is a task which can be solved only by visual input. Often visual task can be solved as a binary classification problem, which is called image classification, but there are also numerous other applications for CV. This chapter will focus on image classification, semantic segmentation and object detection with their usual benchmarks datasets.

2.3.3.2.1 ImageNet Versions

It's not only common to pre-train your model on ImageNet datasets it's also common to benchmark the models on them. There are many different variants of ImageNet. There is ImageNet-R, a version with non-natural images such as art, cartoons and sketches, or ImageNet-A, which is a more challenging version because they use adversarial images (Goodfellow et al., XXXXc), or ImageNet-V2 (Recht et al., XXXX). The last was created to check whether there is an over-fitting on the classic pre-training ImageNet dataset. They followed the creation process of the original dataset and tested to what extent current classification models generalize to new data. Recht et al. (XXXX) found accuracy drops for all models and suggested that these drops are not caused by adaptivity, but by the models' inability to generalize to slightly "harder" images than those found in the original test sets.

The goal of image classification is to classify the image by assigning a label. Typically, Image Classification refers to images in which only one object appears. To assess the performance one mainly uses Top-1 accuracy, the model's answer with highest probability must be exactly the expected answer, or Top-5 accuracy. Top-5 accuracy means that any of five highest probability answers must match the expected answer. Beyer et al. (XXXX) tried to answer the question "Are we done with ImageNet?" in their paper. Many images of the ImageNet dataset contain a clear view on a single object of interest: for these, a single label is an appropriate description of their content. However many other images contain multiple, similarly prominent objects, limiting the relevance of a single label (Beyer et al., XXXX). In these cases, the ImageNet label is just one of many equally valid descriptions of the image and as a result an image classifier can be penalized for producing a correct description that happens to not coincide with that chosen by the ImageNet label.

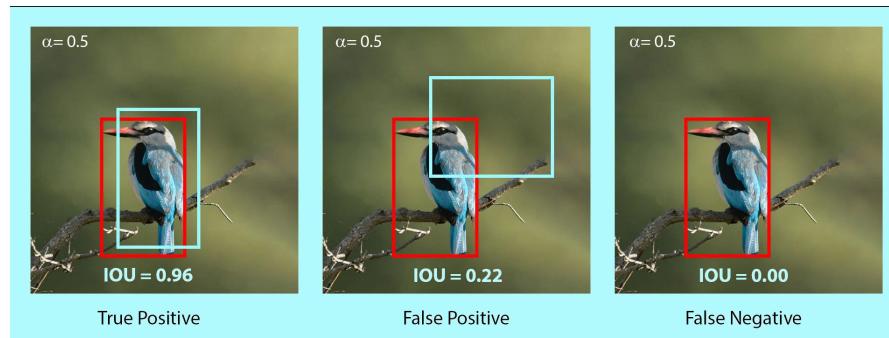
In short a single label per image is not sufficient in many cases. They concluded

yes and no as an answer to the question “Are we done with ImageNet?”. The shortcomings of ImageNet labels and their accuracy were identified and they provided a new ImageNet validation set ReaL (Beyer et al., XXXX) (“Reassessed Labels”) and also a new metric, called ReaL accuracy (Beyer et al., XXXX). The ReaL accuracy measures the precision of the model’s top-1 prediction, which is deemed correct if it is included in the set of labels. These findings suggested that although the original set of labels may be nearing the end of their useful life, ImageNet and its ReaL labels can readily benchmark progress in visual recognition for the foreseeable future.

An addition of a localization tasks to the classification tasks results into object detection. It is used to analyze more realistic cases, like mentioned above, in which multiple objects may or may not exist in an image. The location of an object is typically represented by a bounding box.

2.3.3.2.2 MS-COCO & Object365

In the recent years, the Microsoft COCO dataset or the Object365 data have become the standards to evaluate object detection algorithms, but it’s also possible to use a ImageNet dataset. The primary challenge metric is called mean Average Precision (mAP) at Intersection over Union (IoU) = .50:.05:.95. The IoU is the intersection of the predicted and ground truth boxes divided by the union of the predicted and ground truth boxes. IoU, also called Jaccard Index, values range from 0 to 1. Where 0 means no overlap and 1 means perfect overlap. But how is precision captured in the context of object detection? Precision is known as the ratio of *True Positive*/(*True Positive* + *False Positive*). With the help of the IoU threshold, it’s possible to decide whether the prediction is True Positive(TP), False Positive(FP), or False Negative(FN). The example below shows predictions with IoU threshold set at 0.5.



The .50:.05:.95 means that one uses 10 IoU thresholds of {0.50, 0.55, 0.60, ..., 0.95}. COCO uses this as primary metric, because it rewards detectors with better localization (Microsoft, XXXX).

Object detection and image segmentation are both tasks which are concerned

with localizing objects of interest in an image, but in contrast to object detection image segmentation focuses on pixel-level grouping of different semantics.

Image segmentation can be splitted into various tasks including instance segmentation, panoptic segmentation, and semantic segmentation. Instance segmentation is a task that requires the identification and segmentation of individual instance in an image. Semantic segmentation is a task that requires segmenting all the pixels in the image based on their class label. Panoptic segmentation is a combination of semantic and instance segmentation. The task is to classify all the pixels belonging to a class label, but also identify what instance of class they belong to. Panoptic and instance segmentation is often done on COCO.

2.3.3.2.3 ADE20k

Semantic segmentation can be done one ADE20K(?). ADE are the first three letters of the name Adela Barriuso, who single handedly annotated the entire dataset and 20K is a reference to being roughly 20,000 images in the dataset. This dataset shows a high annotation complexity, because any image in ADE20K contains at least five objects, and the maximum number of object instances per image reaches 273. To asses the performance of a model on the ADE20K dataset one uses the mean IoU. It indicates the IoU between the predicted and ground-truth pixels, averaged over all the classes.

In contrast to the object detection task, the definition of TP, FP, and FN is slightly different as it is not based on a predefined threshold. TP is now the area of intersection between Ground Truth and segmentation mask. FP is the predicted area outside the Ground Truth. FN is the number of pixels in the Ground Truth area that the model failed to predict. The calculation of IoU is the same as in object detection tasks. It's the intersection of the predicted and ground truth boxes aka. TP divided by the union of the predicted and ground truth boxes, which is essentially $TP + FN + FP$. A example is shown down below.



FIGURE 2.9: taken from <https://learnopencv.com>

2.3.3.3 Multi-Modal Benchmarks

Visual understanding goes well beyond object recognition or semantic segmentation. With one glance at an image, a human can effortlessly imagine the world beyond the pixels. This is emphasized by the quote “a picture says more than a thousand words”. High-order of cognition and commonsense reasoning about the world is required to infer people’s actions, goals, and mental states. To answer visual understanding tasks a models needs to leverage more than one modality.

2.3.3.3.1 Visual Commonsense Reasoning (VCR)

Visual understanding tasks require seamless integration between recognition and cognition and this task can be formalized as Visual Commonsense Reasoning (VCR). [Zellers et al. \(XXXXa\)](#) introduce a new dataset called VCR. It consists of 290k multiple choice QA problems derived from 110k movie scenes. The key recipe for generating non-trivial and high-quality problems at scale is Adversarial Matching. Incorrect choices are obtained via maximum-weight bipartite matching between queries and responses. This matching transforms rich annotations into multiple choice questions with minimal bias. VCR casted as a four-way multiple choice task.

The underlying scenes come from the Large Scale Movie Description Challenge and YouTube movie clips and they searched for interesting and diverse situations to ensure this they trained and applied an “interestingnes filter”. The most interesting images were passed to Workers of Amazon Mechanical Turk. Additional context in form of video caption was given to the worker. After reading this they had to propose one to three questions about the image. For each question, they had to provide a reasonable answer and a rationale. This results is an underlying dataset with high agreement and diversity of reasoning. Almost every answer and rationale is unique. To make these cognition-level questions simple to ask, and to avoid the clunkiness of referring expressions, VCR’s language integrates object tags ([person2]) and explicitly excludes referring expressions (‘the woman on the right.’). These object tags are detected from Mask-RCNN. The following types of questions are in the benchmarks: 38% Explanation (‘Why is [person1] wearing sunglasses inside?’), 24% Activity (‘What are [person1] and person[2] doing;’), 13% Temporal (“What will [person6] do after unpacking the groceries?”), 8% Mental, 7% Role, 5% Scene, 5% Hypothetical.

So in this setup, a model is provided a question, and has to pick the best answer out of four choices. Only one of the four is correct. If the model answered correctly a new question, along with the correct answer, is provided. Now the model has to justify it by picking the best rationale out of four choices. The first part is called Question Answering ($Q \rightarrow A$) and the second part Answer Justification ($QA \rightarrow R$). They combine both parts into a $Q \rightarrow AR$ metric, in

which a model only gets a question right if it answers correctly and picks the right rationale. If it gets either the answer or the rationale wrong, the entire prediction will be wrong. Models are evaluated in terms of accuracy.

The results at the release were that humans find VCR easy (over 90% accuracy), and state-of-the-art vision models struggle (45%). At the moment of writing, the best model achieves 85.5 in ($Q \rightarrow A$), 87.5 in ($QA \rightarrow R$) and 74.9 in $Q \rightarrow AR$. So the models are closing the gap but VCR is still far from solved. An “simpler” approach to evaluate vision-language models is to ask questions without reasoning about an image.

2.3.3.3.2 Visual Question Answering 1.0 & 2.0 (VQA)

For this reason Antol et al. (XXXX) created an open-ended answering task and a multiple-choice task. Their dataset contains roughly 250k images, 760k questions, and 10M answers. 204k images are taken from the MS COCO dataset but also newly created datasets are used. Three questions were collected for each image or scene. Each question was answered by ten subjects along with their confidence. The dataset contains over 760K questions with around 10M answers. “what”-, “how”-, “is”- questions are mainly used in the benchmark. But they had major flaws in their creation. A model which blindly answering “yes” without reading the rest of the question or looking at the associated image results in a VQA accuracy of 87% or the most common sport answer “tennis” was the correct answer for 41% of the questions starting with “What sport is”, and “2” is the correct answer for 39% of the questions starting with “How many” (Antol et al., XXXX).

Zhang et al. (XXXXa) pointed out a particular ‘visual priming bias’ in the VQA dataset. Zhang et al. (XXXXa) showed that language provides a strong prior that can result in good superficial performance, without the underlying models truly understanding the visual content. Zhang et al. (XXXXa) collected a balanced dataset containing pairs of complementary scenes to reduce or eliminate the strong prior of the language. Goyal et al. (XXXX) did the same and made a second iteration of the Visual Question Answering Dataset and Challenge (VQA v2.0). Goyal et al. (XXXX) balanced the popular VQA dataset (Antol et al., XXXX) by collecting complementary images such that every question in balanced dataset is associated with not just a single image, but rather a pair of similar images that result in two different answers to the question. The dataset is by construction more balanced than the original VQA dataset and has approximately twice the number of image-question pairs.

2.3.3.4 GQA

Hudson and Manning (XXXX) introduced the GQA dataset for real-world visual reasoning and compositional question answering. It consists of 113K

images and 22M questions of assorted types and varying compositionality degrees, measuring performance on an array of reasoning skills such as object and attribute recognition, transitive relation tracking, spatial reasoning, logical inference and comparisons. They also proposed Consistency, Validity and Plausibility as new measures to get more insight into models' behavior and performance. Consistency measures responses consistency across different questions. To achieve a high consistency a model may require deeper understanding of the question semantics in context of the image. The validity metric checks whether a given answer is in the question scope, e.g. responding some color to a color question. The plausibility score goes a step further, measuring whether the answer is reasonable, or makes sense, given the question (e.g. elephant usually do not eat pizza).

They even made a comparison between GQA and VQA 2.0. They came to the conclusion that the questions of GQA are objective, unambiguous, more compositional and can be answered from the images only, potentially making this benchmark more controlled and convenient for making research progress on. Conversely, VQA questions tend to be a bit more ambiguous and subjective, at times with no clear and conclusive answer. Finally, we can see that GQA provides more questions for each image and thus covers it more thoroughly than VQA.

2.3.3.4.1 Generative Benchmarks

Almost everybody is talking right now about generative models like DALL-E2, Imagen, Parti. It seems like every month a new one is presented. But how can we compare these models? Automatic image quality and automatic image-text alignment are two reasonable evaluation metrics. Fréchet Inception Distance (FID) can be used as primary automated metric for measuring image quality. The Frechet Inception Distance compares the distribution of generated images with the distribution of real images that were used to train the generator. A small value is wanted, as it's a distance measure. Text-image fit can be captured through automated captioning evaluation. For this an image output by the model is captioned with a model, which is able to do image captioning. The similarity of the input prompt and the generated caption is then assessed via BLEU, CIDEr, METEOR and SPICE and also human evaluation is done. Here different generative models are used with the same prompts and the human is asked to choose which output is a higher quality image and which is a better match to the input prompt. One always has to keep in mind, that the images of the generative models are always "cherry picked". They do not typically represent, for example, a single shot interaction in which the model directly produces such an image. To make this clear, Yu et al. (XXXX) showed their way of growing the cherry tree.

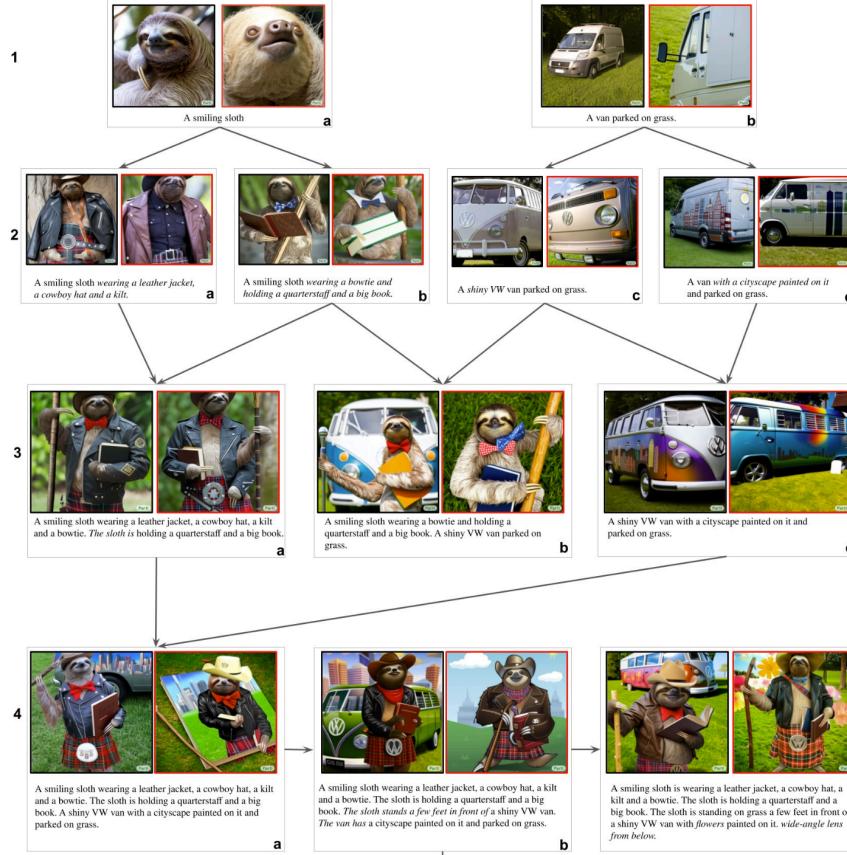


FIGURE 2.10: taken from Parti Paper

2.3.3.4.2 PartiPrompts, DrawBench, Localized Narratives

In a sense, this is a form of model whispering as one stretches such models to their limits. Besides to that they also present PartiPrompts (P2) which is a set of over 1600 (English) prompts curated to measure model capabilities across a variety of categories and controlled dimensions of difficulty. P2 prompts can be simple, but can also be complex, such as 67-word description they created for Vincent van Gogh's The Starry Night. DrawBench is a similar dataset. Also the Localized Narratives dataset from the dataset section consists of long prompts and though it can also be used as a benchmark for generative models.

Current benchmarks give a good perspective on model performance on a wide range of V&L tasks, but the field is only starting to assess why models perform

so well and whether models learn specific capabilities that span multiple V&L tasks.

2.3.3.4.3 FOIL it!

Shekhar et al. (XXXX) proposed an automatic method for creating a large dataset of real images with minimal language bias and some diagnostic abilities. They extended the MS-COCO dataset and created FOIL-COCO. FOIL stands for “Find One mismatch between Image and Language caption” and consists of images associated with incorrect captions. The captions are produced by introducing one single error (or ‘foil’) per caption in existing, human-annotated data. So each datapoint FOIL-COCO can be described as triplet consisting of an image, original and foil caption. Their data generation process consists of four main steps:

1. Generation of replacement word pairs
2. Splitting of replacement pairs into training and testing
3. Generation of foil captions
4. Mining the hardest foil caption for each image

The models are evaluated on three different tasks. The first one is Correct vs. foil classification. Given an image and a caption, the model is asked to mark whether the caption is correct or wrong. The aim is to understand whether LaVi models can spot mismatches between their coarse representations of language and visual input. The second task is Foil word detection. Given an image and a foil caption, the model has to detect the foil word. The aim is to evaluate the understanding of the system at the word level. The last task Foil word correction. Given an image, a foil caption and the foil word, the model has to detect the foil and provide its correction. The aim is to check whether the system’s visual representation is fine-grained enough to be able to extract the information necessary to correct the error. Their hypothesis is that systems which, like humans, deeply integrate the language and vision modalities, should spot foil captions quite easily.

2.3.3.4.4 FALSE

Vision And Language Structured Evaluation (FALSE) (Parcalabescu et al., XXXX) builds on the same idea. This benchmark aims to gauge the sensitivity of pre-trained V&L models to foiled instances. They coverd a wide spectrum of basic linguistic phenomena affecting the linguistic and visual modalities: existence, plurality, counting, spatial relations, actions, and entity coreference. To generate the foils they first use strong language models to propose foil and second they use natural language inference to filter out captions that still can describe the image. To do this in an automatic fashion they use the image as

an premise and the caption its entailed hypothesis. Additionally they use the caption as an premise and the foil as the hypothesis. If an NLI model predicts the foil to be neutral or a contradiction with respect to the caption, they see this as an indicator for a good foil. At last the used human annotators validate all generated testing data. Mainly the MS-COCO dataset is used. VALSE is as a task-independent, zero-shot benchmark to assess the extent to which models learn to ground specific linguistic phenomena as a consequence of their pretraining.

2.3.3.5 Other Benchmarks

As we don't live in a world with unlimited resources, it's also important to keep track of how much energy is consumed to train the models and how big the carbon footprint is. Strubell et al. (XXXXa) investigated some NLP models and benchmarked model training and development costs in terms of dollars and estimated CO_2 emissions. They came to the result that training a single BERT base model without hyperparameter tuning on GPUs requires the same energy as a trans-American flight. On average a human is responsible for 5t CO_2 per year and Strubell et al. (XXXXa) estimated that the training procedure of a big Transformer with neural architecture search emitted 284t of CO_2 . Works (Lottick et al., XXXX, Henderson et al. (XXXX)) have released online tools to benchmark their energy usage and initiatives such as the SustainNLP workshop have since taken up the goal of prioritizing computationally efficient hardware and algorithms. These findings are just some points one should keep in mind.

In the following chapters we will see how the multimodal architectures use these datasets and also how they perform on the given benchmarks.



3

Multimodal architectures

Authors: Luyang Chu, Karol Urbanczyk, Giacomo Loss, Max Schneider, Stefan Jauch-Walser

Supervisor: Christian Heumann

Multimodal learning refers to the process of learning representations from different types of input modalities, such as image data, text or speech. Due to methodological breakthroughs in the fields of Natural Language Processing (NLP) as well as Computer Vision (CV), in recent years multimodal models have gained increasing attention as they are able to strengthen predictions and better emulate the way humans learn. This chapter focuses on discussing images and text as input data. The remainder of the chapter is structured as follows:

The first part “Image2Text” discusses how transformer-based architectures improve meaningful captioning for complex images using a new large scale, richly annotated dataset COCO ([Lin et al., XXXXc; ?](#)). Whether it is seeing a photograph and describing it or parsing a complex scene and describing its context, it is not a difficult task for humans. But it is much more complex and challenging for computers. We start with focusing on images as input modalities. In 2014 Microsoft COCO was developed with a primary goal of advancing the state-of-the-art (SOTA) in object recognition by diving deeper into a broader question of scene understanding ([Lin et al., XXXXc](#)). COCO stands for Common Objects in Context. It addresses three core problems in scene understanding: object detection (non-iconic views), segmentation, and captioning. For tasks like machine translation and language understanding in NLP, transformer-based architecture is widely used. However, the potential of these applications in the multi-modal context has not been fully covered. With the help of the COCO dataset, a transformer-based architecture: Meshed-Memory Transformer for Image Captioning (M^2) will be introduced to improve both image encoding and the language generation steps (?). The performance of the (M^2) Transformer and different fully-attentive models will be evaluated and compared on the COCO dataset.

Next, in “Text2Image”, the idea of incorporating textual input in order to generate visual representations is described. Current advancements in this field have been made possible largely due to recent breakthroughs in NLP, which first allowed for learning contextual representations of text. Transformer-like

architectures are being used to encode the input into embedding vectors, which are later helpful in guiding the process of image generation. The chapter looks into details and discusses two SOTA model architectures by OpenAI, which both condition on text representations. Surprisingly, none of them uses a GAN approach - a method which probably has been seen as the go-to idea for image generation over the last years. The first model is DALL-E (?), which essentially combines Variational Encoder (VAE) with Autoregressive Transformer. In the first step, VAE is being trained to learn downsized image representations. Such embeddings are concatenated with text embeddings into one text-image pair input. However, both of them use different dimensionality and vocabulary size. In the second step, the transformer is trained on a next token prediction task given these data pairs. Finally, at inference time, the model is able to generate images in the following way:

1. Encode text input into text embedding
2. Use trained transformer from step 2 to generate image embedding
3. Use VAE from step 1 to generate image from image embedding

The next approach to text-to-image generation is a GLIDE model (?). GLIDE stands for Guided Language to Image Diffusion for Generation and Editing. Its idea is to use Diffusion Models. In its core, Diffusion Model is a simple idea – random noise is being added to the image in an iterative fashion, and then model learns how to reconstruct this image. In the case of GLIDE this learning process is conditioned on the text prompt, which is first passed through a transformer. Both models differ in their results. While DALL-E’s resulting images might have been overwhelming back in the beginning of 2021, GLIDE is thought to significantly improve on photorealism and resolution the generated images. Since the field has already seen further improvements following GLIDE, these new developments are also going to be mentioned in the chapter.

The third part, “Images supporting Language Models”, deals with the integration of visual elements in pure textual language models. Distributional semantic models such as Word2Vec and BERT assume that the meaning of a given word or sentence can be understood by looking at how (in which context) and when the word or the sentence appear in the text corpus, namely from its “distribution” within the text. But this assumption has been historically questioned, because words and sentences must be grounded in other perceptual dimensions in order to understand their meaning (see for example the “symbol grounding problem”; Harnad, XXXX). For these reasons, a broad range of models has been developed with the aim to improve pure language models, leveraging on the addition of other perceptual information, such as visual ones. This subchapter focuses in particular on the integration of visual elements (images) to support pure language models for various tasks at the word-level and sentence-level. The starting point is always a language model, on which visual representations (extracted often with the help of large pools of images like MS COCO, see chapter “Img2Text” for further references) are

to be “integrated”. But how? There has been proposed a wide range of solutions: On one side of the spectrum, textual elements and visual ones are learned separately and then “combined” together whereas on the other side, the learning of textual and visual features takes place simultaneously/jointly.

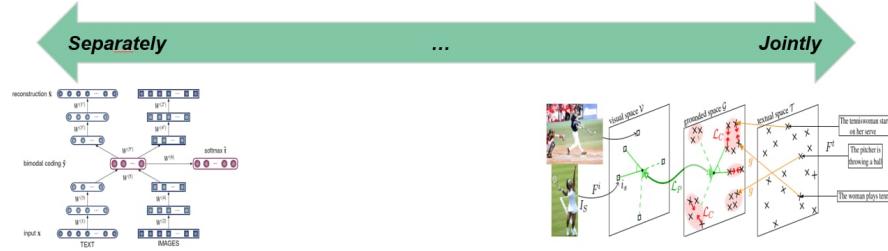


FIGURE 3.1: Left, Silberer et al., 2014: stacked autoencoders to learn higher-level embeddings from textual and visual modalities, encoded as vectors of attributes. Right, Bordes et al., 2020: textual and visual information fused in an Intermediate space denoted as “grounded space”; the “grounding objective function” is not applied directly on sentence embeddings but trained on this intermediate space, on which sentence embeddings are projected.

For example, [Silberer and Lapata \(XXXXa\)](#) implement a model where a one-to-one correspondence between textual and visual space is assumed. Text and visual representations are passed to two separate unimodal encoders and both outputs are then fed to a bimodal autoencoder. On the other side, [?](#) propose a “text objective function” whose parameters are shared with an additional “grounded objective function”. The training of the latter takes place in what the authors called a “grounded space”, which allows to avoid the one-to-one correspondence between textual and visual space. These are just introductory examples and between these two approaches there are many shades of gray (maybe more than fifty...). These models exhibit in many instances better performance than pure language models, but they still struggle on some aspects, for example when they deal with abstract words and sentences.

Afterwards, in “Text supporting Image Models”, approaches where natural language is used as supervision for CV models are described. Intuitively these models should be more powerful compared to models supervised solely by manually labeled data, simply because there is much more training data available.

An important example for this is the CLIP model ([Radford et al., XXXXa](#)) with its new dataset WIT (WebImageText) comprising 400 million text-image pairs scraped from the internet. Similar to “Text2Image” the recent successes in NLP have inspired new approaches in this field. Most importantly pre-train methods, which directly learn from raw text (e. g. GPT-n, Generative Pre-trained Transformer; [Brown et al., XXXXa](#)). So, CLIP stands for Con-

trastive Language-Image Pre-training. A transformer-like architecture is used for jointly pre-training a text encoder and an image encoder. For this the contrastive goal to correctly predict which natural language text pertains to which image inside a certain batch, is employed. Training this way turned out to be more efficient than to generate captions for images. This leads to a flexible model, which at test time uses the learned text encoder as a “zero-shot” classifier on embeddings of the target dataset’s classes. The model, for example, can perform optical character recognition, geo-location and action-recognition. Performance-wise CLIP can be competitive with task-specific supervised models, while never seeing an instance of the specific dataset before. This suggests an important step towards closing the “robustness gap”, where machine learning models fail to meet the expectations set by their previous performance – especially on ImageNet test-sets – on new datasets.

Finally, “Text plus Images” discusses how text and image inputs can be incorporated into a single unifying framework in order to get closer to a general self-supervised learning model. There are two key advantages that make such a model particularly interesting. Similar to models mentioned in previous parts, devoid of human labelling, self-supervised models don’t suffer from the same capacity constraints as regular supervised learning models. On top of that, while there have been notable advances in dealing with different modalities using single modality models, it is often unclear to which extend a model structure generalizes across different modalities. Rather than potentially learning modality-specific biases, a general multipurpose framework can help increase robustness while also simplifying the learner portfolio. In order to investigate different challenges and trends in vision-and-language modelling, this section takes a closer look at three different models, namely data2vec ([Baevski et al. \(XXXXa\)](#)), VilBert ([Lu et al. \(XXXXb\)](#)) and Flamingo ([Alayrac et al. \(XXXXa\)](#)). Data2vec is a new multimodal self-supervised learning model which uses a single framework for either speech, natural language processing or computer vision. This is in contrast to earlier models which used different algorithms for different modalities. The core idea of data2vec, developed by MetaAI, is to predict latent representations of the full input data based on a masked view of the input in a self-distillation setup using a standard transformer architecture. ([Baevski et al. \(XXXXa\)](#)) As a result, the main improvement is in the framework itself, not the underlying models themselves. For example, the transformer architecture being used follows [Vaswani et al. \(XXXXb\)](#). Through their parallelizability, transformers have several advantages over CNNs particularly when large amounts of data are being used, making them the standard approach in vision-language modelling. ([Dosovitskiy et al. \(XXXX\)](#)) VilBert is an earlier model that in contrast to data2vec can handle cross-modality tasks. Finally, Flamingo is a modern few shot learning model which features 80B parameters - significantly more than the other two models. Through a large language model incorporated in its architecture, it has great text generating capabilities to tackle open-ended tasks. It also poses the question how to efficiently train increasingly large

models and shows the effectiveness of using perceiver architectures ([Jaegle et al. \(XXXX\)](#)) to encode inputs from different modalities as well as how to leverage communication between pretrained and frozen models.

3.1 Img2text

*Author: Luyang Chu

*Supervisor: Christian Heumann

Image captioning means autonomously producing descriptive sentences for images. It has stimulated interest in both natural language processing and computer vision research in recent years. Image captioning is a key task that requires a semantic comprehension of images as well as the capacity to generate accurate and precise description sentences.

3.1.1 Microsoft COCO: Common Objects in Context

Understanding of visual scenes plays an important role in computer vision research (CV). Many tasks are included in it, such as image classification, object detection, object localization and semantic scene labeling. Through the computer vision research history, Image Datasets have played a critical role. They are not only essential for training and evaluating new algorithms, but also lead the research to new challenging directions. ([Lin et al., XXXXc](#)) In the early year, researchers developed Datasets ([Deng et al., XXXX](#)), ([Xiao et al., XXXX](#)), ([Everingham et al., XXXX](#)) which enabled the direct comparison of hundreds of image recognition algorithms, that was the early evolution in object recognition. Recent years, ImageNet ([Deng et al., XXXX](#)) which contains millions of images has enabled breakthroughs in both object classification and detection research using new deep learning algorithms.

With the goal of advancing the state-of-art in object recognition especially scene understanding, a new large scale data called Microsoft COCO was published in 2014. MS COCO focuses on three core problems in scene understanding: detecting non-iconic views, detecting the semantic relationships between objects and precise localization of image objects. ([Lin et al., XXXXc](#))

MS COCO Dataset contains 91 common object categories with a total of 328,000 images as well as 2,500,000 labeled instances. All these images could be recognized by a 4 year old child. 82 categories include more than 5000 labeled The labeled instances which may support the detection of relationships between objects in COCO. In order to provide precise localization of object

instances, only “Thing” categories like car, table, dog will be included. Objects which do not have clear boundaries like sky, sea, grass, will not be included. In current object recognition research, algorithms perform well on images with iconic views. These images always contains the single object category in the center of the image. To accomplish the goal of detecting the contextual relationships between objects, more complex images with multiple objects or natural images which comes from our daily life are gathered for the Dataset.

In addition to MC COCO, researchers have been working on the development of new large databases. In recent years many new large databases like ImageNet, PASCAL VOC and SUN have been developed in the field of computer vision. Each of this dataset has its own specific focus.

Datasets for object recognition can be roughly split into three groups: object classification, object detection and semantic scene labeling.

Object classification requires binary labels to indicate whether objects are present in an image, ImageNet ([Deng et al., XXXX](#)) is clearly distinguishable from other datasets in terms of the dataset sizes. ImageNet is containing 22k categories with 500-1000 images each. ([Lin et al., XXXXc](#)), in comparison to other datasets, the ImageNet dataset contains over 14 million labeled images with both entry-level and fine-grained categories by using WordNet Hierarchy and has enabled significant advances in image classification.

Detecting an object includes two steps, first is to ensure that an object from a specified class is present, the second step is to localize the object in the image with a bounding box. This can be implemented into tasks like face detection or pedestrians detection. The PASCAL VOC ([Everingham et al., XXXX](#)) dataset is used to help the detection of basic object categories. With 20 object categories and over 11,000 images, PASCAL VOC labeled over 27,000 object instance by using bounding boxes. Almost 7,000 object instances from them had detailed segmentations. ([Lin et al., XXXXc](#))

Labeling semantic objects in a scene requires that each pixel of an image be labeled as belonging to a category, such as sky, chair, etc, but individual instances of objects do not need to be segmented. ([Lin et al., XXXXc](#)) Some objects like sky, grass, street can also be defined and labeled in this way. The dataset SUN ([Xiao et al., XXXX](#)) combines many of the properties of both object detection and semantic scene labeling datasets for the task of scene understanding, it contains 908 scene categories from the WordNet dictionary ([Fellbaum, XXXX](#)) with segmented objects. The 3,819 object categories split them to object detection datasets (person, chair) and to semantic scene labeling (wall, sky, floor). ([Lin et al., XXXXc](#))

3.1.1.1 Image Collection and Annotation for MS COCO

COCO is a large-scale richly annotated Datatset, the progress of building consists of two phases: Data collection and image annotation.

In order to select representative object categories for images in COCO, researchers collected several categories from different dataset like PASCAL VOC ([Everingham et al., XXXX](#)) and other sources. All these object categories can be recognized by children between 4 to 8. The quality of the object categories were ensured by co-authors. Co-authors scale the categories from 1 to 5 depending on their common occurrence, practical applications and diversity from other categories. ([Lin et al., XXXXc](#)) The final number of the list is 91. All the categories from PASCAL VOC are also included in COCO. ([Everingham et al., XXXX](#))

With the help of representative object categories, COCO wants to collect a dataset which a majority of these images are non-iconic. All these images can be roughly divided into three types in Fig 3.2 :iconic-object images, iconic-scene images and non-iconic images. ([Lin et al., XXXXc](#))



Fig. 2: Example of (a) iconic object images, (b) iconic scene images, and (c) non-iconic images.

FIGURE 3.2: Type of images ([Lin et al., XXXXc](#)).

Images are collected through two strategies, firstly images from Flickr which contains photos uploaded by amateur photographer with keywords are collected. Secondly, researchers will search for pairwise combination of object categories like “dog + car” to gather more non-iconic images and images with rich contextual relationships. ([Lin et al., XXXXc](#))

Due to the the scale of the dataset and the high cost of the annotation process, the design of a high quality annotation pipeline with efficient cost is a difficult task. The annotation pipeline in Fig 3.3 for COCO is splitted into three primary tasks: 1. category labeling, 2.instance spotting, and 3. instance segmenting. ([Lin et al., XXXXc](#))

As we can see in the Fig 3.3, object categories in each image will be determined in the first step. Due to the large number of Datasets and categories, they used a hierarchical approach instead of doing binary classification for each category. All the 91 categories have divided into 11 super-categories. The worker will examine the existence of a single instance for a given super-category. Workers will only label one instance for each super-categories with an category’s icon. ([Lin et al., XXXXc](#)) For each image, 8 workers were asked to label it. This

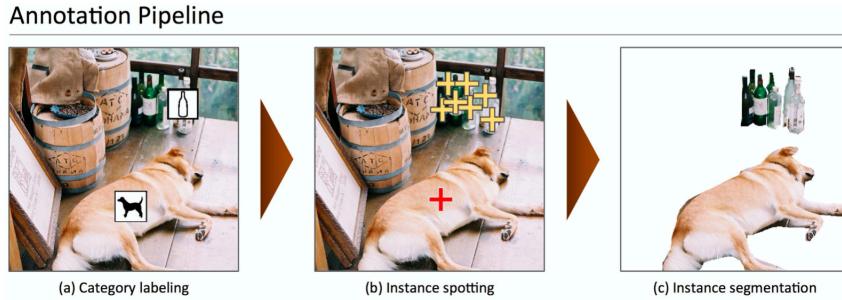


FIGURE 3.3: Annotation pipeline for COCO (Lin et al., XXXXc).

hierarchical approach has helped to reduce the time for labeling. However, the first phase still took 20k worker hours to complete.(Lin et al., XXXXc)

In the next step, all instances of the object categories in an image were labeled, at most 10 instances of a given category per image will be labeled by each worker. In both instance spotting and instance segmenting steps, the location of the instance found by a worker in the previous stage can be seen by the current worker. Each image was labeled by 8 workers for a total of 10k worker hours.(Lin et al., XXXXc)

In the final segmenting stage, each object instance is segmented, the segmentation for other instances and the specification of the object instance by a worker in the previous stage will also shown to the worker.Segmenting 2.5 Mio object instances is an extremely time consuming task which requires over 22 worker hours per 1,000 segmentations.To minimize cost and improve the quality of segmentation, all workers are required to complete a training task for each object category. In order to ensure a better quality, an explicit verification step on each segmented instance was performed.

3.1.1.2 Comparison with other Datasets

In recent years, researchers have developed several pre-trained datasets and benchmarks which helped the development of Algorithms for CV. Each of these datasets varies significantly in size, list of labeled categories and types of images. In the previous part we also introduced the different research main focus of some Datasets like ImageNet (Deng et al., XXXX), PASCAL VOC (Everingham et al., XXXX) and SUN (Xiao et al., XXXX) ImageNet containing millions of images has enabled breakthroughs in both object classification and detection research using a new class of deep learning algorithms. It was created to capture a large number of object categories, many of which are fine-grained. SUN focuses on labeling scene types and the objects that commonly occur in them. Finally, PASCAL VOC's primary application is object detection in

natural images. MS COCO is designed for the detection and segmentation of objects occurring in their natural context.(Lin et al., XXXXc)

With the help of Fig 3.4, we could compare COCO with ImageNet PASCAL VOC and SUN from different aspects.(Lin et al., XXXXc)

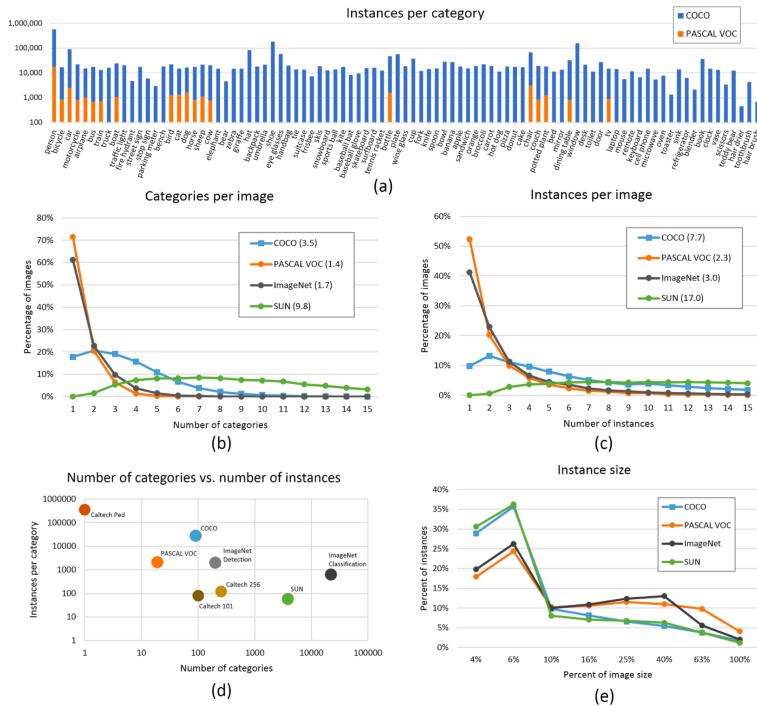


FIGURE 3.4: Comparison COCO with other PASCAL VOC, SUN and ImageNet (Lin et al., XXXXc).

The number of instances per category for all 91 categories for COCO and PASCAL VOC is shown in (a). Compare to PASCAL VOC, COCO has both more categories and more instances per categories.The number of object categories and the number of instances per category for all the datasets is shown in (d). MS COCO has fewer categories than ImageNet and SUN, but it has the most instances per category among all the dataset, which from the hypothesis of researchers might be useful for learning complex models capable of precise localization.(Lin et al., XXXXc) (b,c) show the number of annotated categories and annotated instances per image for MS COCO, ImageNet Detection, PASCAL VOC and SUN (average number of categories and instances are shown in parentheses). On average COCO contains 3.5 categories and 7.7 instances per image. ImageNet and PASCAL VOC both have less than 2 categories and 3 instances per image on average. SUN dataset has the most contextual

information, 9,8 categories and 17 instances per image. (e) is the distribution of instance sizes for the MS COCO, ImageNet Detection, PASCAL VOC and SUN dataset.(Lin et al., XXXXc)

3.1.1.3 Discussion

COCO is a new large scale data set for detecting and segmenting objects found in everyday life, with the aim of the state-of-the-art in object recognition and scene understanding. It focuses on non-iconic images of objects in natural environments and contains rich contextual information with many objects present per image. COCO is typical vision datasets, which are labor intensive and costly to create. With the vast cost and over 70,000 worker hours, 2.5 Mio instances were annotated to drive the advancement of object detection and segmentation algorithms. COCO is a good benchmark for the field of CV.(Lin et al., XXXXc) The COCO Team also shows directions for future. For example “stuff” label like “sky”, “grass” and “street” etc, may also be included in the dataset since “stuff” categories provide significant contextual information for the object detection.

3.1.2 Models for Image captioning

The image captioning task generalizes describe the visual content of an image in natural language, so it requires an algorithm to understand and model the relationships between visual and textual elements, and to generate a sequence of output words.(?) In the last few years, collections of methods have been proposed for image captioning. Earlier approaches based on generations pf simple templates, which contains the output produced from the object detector or attribute predictor. (Socher and Fei-fei, XXXX), (Yao et al., XXXXa). With the sequential nature of language, most research on image captioning has focused on deep learning techniques,especially using Recurrent Neural Network models (RNNs)(Vinyals et al., XXXX),(Karpathy and Fei-Fei, XXXX) or LSTMs. Mostly, RNNs are used as languages models, and visual information and sequence generation will be encoded from the output of a CNN. With the aim of modelling the relationships between image regions, words, graph convolution neural network in the image encoding phase (Yao et al., XXXXb) or single-layer attention mechanisms (Xu et al., XXXX) on the image encoding side have been proposed to incorporate more semantic and spatial relationships between objects. RNN-based-models are widely adopted, however, the model has its limitation on representation power and sequential nature.(?) Recently, new fully-attentive models, in which the use of self-attention has replaced the recurrence,have been proposed. New approaches apply the Transformer models (Vaswani et al., XXXXc) and BERT (Devlin et al., XXXXc) models to image captioning tasks. Transformer consists of an encoder with a stack of self-attention and feed-forward layers, and a decoder which uses self-attention

on words and cross-attention over the output of the last encoder layer.(?). In some Transformer based approaches, a Transformer-like encoder was paired with an LSTM decoder. While the aforementioned approaches have exploited the original Transformer architecture. For example Herdade *et al.* ([Herdade et al., XXXX](#)) proposed Transformer architecture for image captioning with the focus on geometric relations between input objects at the same time. Specifically, an additional geometric weight between object pairs which is used to scale attention weights are computed. Similarly, an extension of the attention operator in which the final attended information is weighted by a gate guided by the context was introduced by Huang *et al.* ([Huang et al., XXXX](#)),(?).

3.1.3 Meshed-Memory Transformer for Image Captioning (M^2)

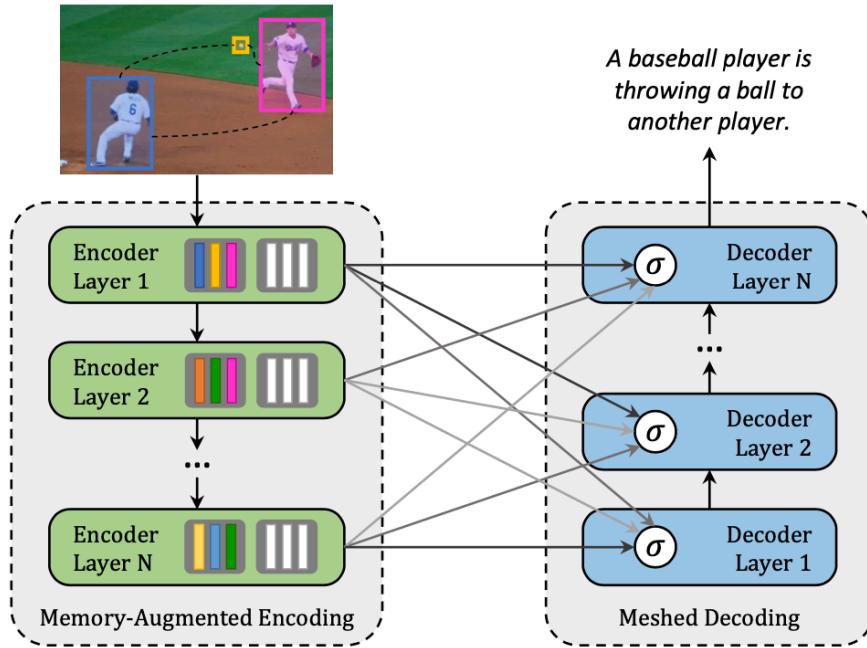
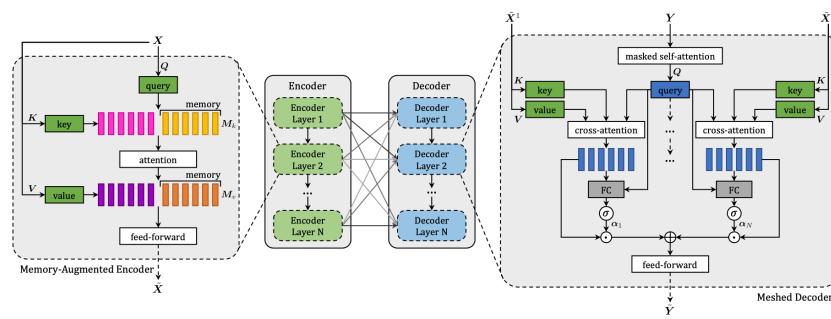
Although Transformer-based architectures have been widely implemented in sequence modeling tasks like machine translation and language understanding. However, its applicability for multi-modal tasks like image captioning is still largely under-explored. (?)

A novel fully-attentive approach called Meshed-Memory Transformer for Image Captioning (M^2) for image captioning is proposed by Marcella *et al*(?) with the aim of improving the design of both the image encoder and the language decoder. Compare to all previous image captioning models, M^2 (see Fig. 3.5) has two new novelties, the encoder encodes a multi-level representation of the relationships between image regions with respect to low-level and high-level relations, a priori knowledge can be learned and modeled by using persistent memory vectors. The multi-layer architecture exploits both low- and high-level visual relationships through a learned gating mechanism, which compute the weight at each level, therefore, a mesh-like connectivity between encoder and decoder layers is created for sentence generation process.(?)

3.1.3.1 M^2 Transformer Architecture

Fig 3.6 shows the detailed architecture of M^2 Transformer. M^2 Transformer can be divided into Encoder (left) module and Decoder (right) module, both module has multiple layers. Given the input Image region X , it go through the Attention and feed forward layer. The relationship between image regions with a priori knowledge will be encoded in each encoding layer, the output of each encoding layers will be read by decoding layers to generate the caption for image word by word.(?)

All interactions between word and image-level features of the input image X are modeled by using scaled dot-product attention. Attention operates on vectors of queries Q , keys K and values V , and takes a weighted sum of value

FIGURE 3.5: M^2 Transformer (?).FIGURE 3.6: M^2 Transformer Architecture (?).

vectors according to a similarity distribution between query and key vectors. Attention can be defined as follows (?):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (3.1)$$

where Q is a matrix of n_q query vectors, K and V both contain n_k keys and values, all the vectors has the same dimensionality, and d is a scaling factor.

3.1.3.1.1 Memory-Augmented Encoder

For the given image region X , attention can be used to obtain a permutation invariant encoding of X through the self-attention operations, the operator from the Transformer can be defined as follows (?):

$$S(X) = \text{Attention}(W_qX, W_kX, W_vX) \quad (3.2)$$

In this case, queries, keys, and values are linear projections of the input features, W_q , W_k , W_v are learnable weights, they depend solely on the pairwise similarities between linear projections of the input set X . The self-attention operator encodes the pairwise relationships inside the input. But self-attention also has its limitation: a prior knowledge on relationships between image regions can not be modelled. To overcome the limitation, Marcella *et al*(?) introduce **Memory-Augmented Attention** operator by extending the keys and values with additional prior information, which does not depend on image region X . The additional keys and values are initialized as plain learnable vectors which can be directly updated via SGD. The operator can be defined as (?):

$$\begin{aligned} M_{mem}(X) &= \text{Attention}(W_qX, K, V) \\ K &= [W_kX, M_k] \\ V &= [W_vX, M_v] \end{aligned} \quad (3.3)$$

M_k and M_v are learnable matrices, with n_m rows. $[\cdot, \cdot]$ indicates concatenation. The additional keys and value could help to retrieve a priori knowledge from Input while remaining the queries unchanged.(?)

For the **Encoding Layer**, Memory-augmented operator is d into a Transformer-like layer, the output applied to position-wise feed-forward layer (?):

$$F(X)_i = U\sigma(VX_i + b) + c; \quad (3.4)$$

X_i indicates the i -th vector of the input set, and $F(X)_i$ the i -th vector of

the output. Also, $\sigma(\cdot)$ is the ReLU activation function, V and U are learnable weight matrices, b and c are bias terms.(?)

Each components will be included in a residual connection and a layer norm operation. The complete definition of an encoding layer can be finally written as (?):

$$\begin{aligned} Z &= \text{AddNorm}(M_{mem}(X)) \\ \tilde{X} &= \text{AddNorm}(F(Z)) \end{aligned} \quad (3.5)$$

Finally the **Full Encoder** has multiple encoding layers in sequence, therefore the i -th layer uses the output set computed by layer $i - 1$, higher encoding Layers can exploit and refine relationships identified by previous layers, N encoding layers will produce the output $X = (\tilde{X}^1 \dots \tilde{X}^n)$.(?)

3.1.3.1.2 Meshed Decoder

The decoder depends on both previously generated words and image region encodings. **Meshed Cross-Attention** can take the advantage of all the encoding layers to generate sentences for the image. On the right side of the Fig 3.6 shows the structure of the Meshed Decoder. The input sequence vector Y and the output from all encoding layers \tilde{X} are connected by Meshed Attention operator gated through cross-attentions. The meshed attention operator can be formally defined as (?):

$$M_{mesh}(\tilde{X}, Y) = \sum_{i=1}^N \alpha_i C(\tilde{X}^i, Y) \quad (3.6)$$

$C(\cdot, \cdot)$ stands for the encoder-decoder cross-attention, it could be defined with queries from decoder and the keys and values from encoder.(?)

$$C(\tilde{X}^i, Y) = \text{Attention}(W_q Y, W_k \tilde{X}^i, W_v \tilde{X}^i) \quad (3.7)$$

α_i is a matrix of weights same size as the cross-attention results, α_i models both single contribution of each encoding layer and the relative importance between different layers.(?)

$$\alpha_i = \sigma(W_i[Y, C(\tilde{X}^i, Y)] + b_i) \quad (3.8)$$

The $[\cdot, \cdot]$ indicates concatenation and σ is the sigmoid activation function, W_i is a weight matrix, and b_i is a learnable bias vector.(?)

In decoding layers the prediction of a word should only depend on the word generated previously, so the decoder layer comprises a masked self- attention operation, the operator build connection between queries derived from

the t -th element of its input sequence Y with keys and values from left subsequence, i.e. $Y_{\leq t}$.

Similar as the Encoding layer, the decoder layer also contains a position-wise feed-forward layer, so the decoder layer can be finally defined as (?):

$$\begin{aligned} Z &= \text{AddNorm}(M_{mesh}(X, \text{AddNorm}(S_{mask}(Y)))) \\ \tilde{Y} &= \text{AddNorm}(F(Z)), \end{aligned} \quad (3.9)$$

S_{mask} indicates a masked self-attention over time.(?) Decoder with multiple decoder layers, takes the input word vectors and the t -th element of its output sequence to make the prediction of a word for $t + 1$, conditioned on th $Y_{\leq t}$. Finally the decoder takes a linear projection and a softmax operation, which encodes a probability over words in the dictionary.(?)

3.1.3.1.3 Comparison with other models on COCO Datasets

The M^2 Transformer was evaluated on COCO (Lin et al., XXXXc). COCO is the most commonly used Test dataset for image captioning. In stead of using the original COCO datasets, Marcella et al(?) follows the split of COCO provided by Karpathy et al (Karpathy and Fei-Fei, XXXX). Karpathy uses 5000 images for validation, 5000 images for testing and the rest for training.

For model evaluation and comparison, standard captioning metrics like BLEU (Papineni et al., XXXX), METEOR (Banerjee and Lavie, XXXX), ROUGE (Lin, XXXX), CIDEr (Vedantam et al., XXXX), and SPICE (Anderson et al., XXXXa) which have been introduced in the second chapter are used.

	B-1	B-4	M	R	C	S
Transformer (w/ 6 layers as in [39])	79.1	36.2	27.7	56.9	121.8	20.9
Transformer (w/ 3 layers)	79.6	36.5	27.8	57.0	123.6	21.1
Transformer (w/ AoA [14])	80.3	38.8	29.0	58.4	129.1	22.7
\mathcal{M}^2 Transformer ^{1-to-1} (w/o mem.)	80.5	38.2	28.9	58.2	128.4	22.2
\mathcal{M}^2 Transformer ^{1-to-1}	80.3	38.2	28.9	58.2	129.2	22.5
\mathcal{M}^2 Transformer (w/o mem.)	80.4	38.3	29.0	58.2	129.4	22.6
\mathcal{M}^2 Transformer (w/ softmax)	80.3	38.4	29.1	58.3	130.3	22.5
\mathcal{M}^2 Transformer	80.8	39.1	29.2	58.6	131.2	22.6

FIGURE 3.7: Comparison with Transformer-based alternatives (?)

Transformer model in original configuration with six layers has been applied

to captioning, researchers speculated that specific architectures are required for captioning, so variations of the original Transformer are compared with M^2 Transformer. Other Variations are Transformer with three layers and “Attention on Attention” (AoA) approach (Huang et al., XXXX) to the attentive layers, both in the encoder and in the decoder. (?) The second part intends to evaluate the importance of the meshed connections between encoder and decoder layers. M^2 Transformer (1 to 1) is a reduced version of original M^2 Transformer, in which one encoder layer only connect to one corresponding decoder layer instead of connect to all the decoding layers. As we can see from the Fig 3.7, the original Transformer has 121.8 CIDEr, compare with it, the reduced version of M^2 Transformer, we can see the improvement already (129.2 CIDEr). With the respect to meshed connectivity, which help to exploit relationships encoded at all layers and weights them with a sigmoid gating, we could observe better improvement in CIDEr from 129.2 to 131.2. Also the role of memory vectors and the softmax gating schema for M^2 Transformer are also included in the table. Without the memory vector lead to the reduction of the performance of nearly 1 CIDEr in both reduced M^2 Transformer and the original M^2 Transformer.(?)

	B-1	B-4	M	R	C	S
SCST [33]	-	34.2	26.7	55.7	114.0	-
Up-Down [4]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [15]	79.1	36.5	27.7	57.3	121.9	21.2
Up-Down+HIP [49]	-	38.2	28.4	58.3	127.2	21.9
GCN-LSTM [48]	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [46]	80.8	38.4	28.4	58.6	127.8	22.1
ORT [13]	80.5	38.6	28.7	58.4	128.3	22.6
AoANet [14]	80.2	38.9	29.2	58.8	129.8	22.4
M^2 Transformer	80.8	39.1	29.2	58.6	131.2	22.6

FIGURE 3.8: Comparison with the state of the art on the “Karpathy” test split, in single-model setting (?)

Fig 3.8 compares the performance of M^2 Transformer with several recent proposals for image captioning. SCST (Rennie et al., XXXX) and Up- Down (Anderson et al., XXXXb), use attention over the grid of features and attention over regions. RFNet (?), which uses a recurrent fusion network to merge different CNN features; GCN-LSTM (Yao et al., XXXXc) uses a Graph CNN to exploit pairwise relationships between image regions; SGAE (Yang et al., XXXX) uses instead auto-encoding scene graphs. The original AoANet

(Huang et al., XXXX) approach uses attention on attention for encoding image regions and an LSTM language model. Finally, the ORT (Herdade et al., XXXX) uses a plain Transformer and weights attention scores in the region encoder with pairwise distances between detections (?).

In Fig 3.8, the M^2 Transformer exceeds all other models on BLEU-4, METEOR, CIDEr. The performance of M^2 Transformer was very close and competitive with SGAE on BLEU-1 and with ORT on SPICE.



FIGURE 3.9: "Examples of captions generated by M^2 Transformer and the original Transformer model, as well as the corresponding ground-truths (?)

The Fig 3.9 shows some examples of captions generated by M^2 Transformer and the original Transformer model, as well as the corresponding ground-truths. According to the selected examples of captions, M^2 Transformer shows the ability to generate more accurate descriptions of the images, and the approaches could detect the more detailed relationships between image regions.(?).

The M^2 Transformer is a new Transformer- based architecture for image captioning. It improves the image encoding by learning a multi-level representation of the relationships between image regions while exploiting a priori knowledge from each encoding layer, and uses a mesh-like connectivity at decoding stage to exploit low- and high-level features at the language generation steps. The results of model evaluation with COCO shows that, the perfor-

mance of M^2 Transformer approach surpasses most of the recent approaches and achieves a new state of the art on COCO.(?).

3.2 Text-2-image

Author: Karol Urbańczyk

Supervisor: Jann Goschenhofer

- introduce the concept in few sentences
- choice of recent break-throughs is subjective, many important ones not mentioned (GAWWN, LAFITE, Make-a-Scene, probably many others)

Intention of this chapter is to grasp how the field of text-2-image modelling has been changing over the recent years. We will start with basic concepts that has been around since 2014 and end with the state-of-the-art approaches, as of August 2022. Since the field is developing in a rapid pace, with breakthrough models being announced every quarter, we are aware this chapter might soon not be fully covering the field. However, we must notice that cutting edge capabilities of these models tend to come from the scale and software engineering tricks. Therefore, we believe that focusing on the core concepts should make this chapter have a universal character.

3.2.1 Seeking objectivity

- Objectivity in comparing generated images is very hard to grasp
- However, there are some most common datasets and measures that are being used
- This subchapter will quickly present them

3.2.1.1 Datasets

- COCO
- CUB
- Oxford 102

3.2.1.2 Measures

- FID (Frechet Inception Distance)

- IS (Inception Score)
- Human evaluations - photorealism / caption similarity

3.2.2 Generative Adversarial Networks

- quick intro focusing on why it is crucial to start from GANs

3.2.2.1 Vanilla GAN for Image Generation

- intro of GAN

3.2.2.2 Conditioning on Text

- how to encode the text and use it in the generation process
- show some results

3.2.2.3 Stacking generators

- intro of StackGAN, show some results

3.2.2.4 Is attention all you need?

- intro of AttGAN, show some results

3.2.2.5 Variational Autoencoder

- Introducing the concept of VAE
- How is it helpful in generating images

3.2.3 Dall-E starting post-GAN era

- Intro: OpenAI, dataset used, not public, etc
- VQ-VAE and dVAE
- Details how it's working. Combining Transformer with VQ-VAE. Training vs inference
- Results and image examples

3.2.4 GLIDE

- Intro

- Diffusion concept
- details how GLIDE is working
- results / scores
- Limitations / strengths & weaknesses

3.2.5 Dall-E 2

- Intro (mention PR move)
- details how it is working
- results / scores
- Limitations / strengths & weaknesses

3.2.6 Imagen

- Intro
- details how it is working
- results / scores
- Limitations / strengths & weaknesses

3.2.7 Parti

- Intro
- details how it is working
- results / scores
- Limitations / strengths & weaknesses

3.2.8 Open-Source Community

- Although most of the recent work comes from OpenAI and Google, there are very interesting directions taken by the open community
- Mentioning the models and quickly what is happening. VQGAN+CLIP, Latent Diffusion models for sure
- Maybe some links for the reader to play with?

3.2.9 Discussion

Mention the following points and why they matter

- potential business use cases
- open vs closed-source (mention dall-e mini)

- copyrights
 - biases
-

3.3 Images supporting language models

*Author: Giacomo Loss

*Supervisor: Matthias Assenmacher

3.3.1 Words In (Non-Symbolic) Contexts

Imagine you were alone in a foreign country, you could not speak the language and the only resource you had were a dictionary in the foreign language. You see a word written on a sign but you cannot understand its meaning. What could you do? One idea would be to open the dictionary and look the word up. The problem is that the word is defined by using other words in the foreign language. As a second step you would thus look these new words up and continue like that in further steps to the “infinity and beyond” (cit. Buzz Lightyear). But even after looking every single word in the dictionary up, you would still not be able to understand the meaning of the word written on the sign. If on that sign, next to the unknown word, something else was instead depicted, for example an image of a fork and a knife, you might speculate that the word indicates something which has to do with food, like a restaurant. And this without explicitly knowing the meaning of the word. This example is inspired by the work of Stevan Harnad, which formulated at the beginning of the 90’s the so called *Symbol Grounding Problem* (Harnad (XXXX)). It asserts that it is not possible to understand the meaning (semantics) of a word by just looking at other words because words are essentially meaningless symbols. It is possible to understand the meaning only if the word is put in a context, a perceptual space, other than that of written language: the word must be *grounded* in non-symbolic representations, like images, for example. Over the past 10 years there has been a whopping development of distributional semantic models (DSMs, henceforth), especially after the Word2vec (Mikolov et al. (XXXX)) revolution. This family of models assumes that the meaning of words and sentences can be inferred by the “distribution” of those words and sentences within a text corpus (the *Distributional Hypothesis* formulated by Harris et al. (XXXX)). But the *Symbol Grounding Problem* mentioned earlier suggests that DSMs do not resemble the way words are learned by humans, which is in multimodal perceptual contexts. For these reasons, models have been developed with the goal to integrate further modalities (like visual ones) in pure language models, assuming that grounding words and sentences

in other perceptual contexts should lead to a better understanding of their semantics and, as a result, to better performance in pure language tasks.

The focus of this subchapter are models which empower pure language models with visual modalities in form of images: their goal is to obtain better semantic representations (in form of embedding vectors) of words. First, a quick recap of the main pure language models will be provided. After that, the historical evolution of the integration of images as visual modalities into pure language models will be discussed: from simple concatenation of textual and visual modalities, to the projection of visual elements in a common grounded space and more recently, the use of Transformers (see figure 3.10). Eventually, a comprehensive evaluation of the different models against benchmarks will be carried out.

Again, the focus is on how to employ visual elements to obtain embeddings able to capture the semantics of words. More concrete applications, such as those in the field of machine translation are out of scope and will be only marginally addressed at the end of the subchapter.

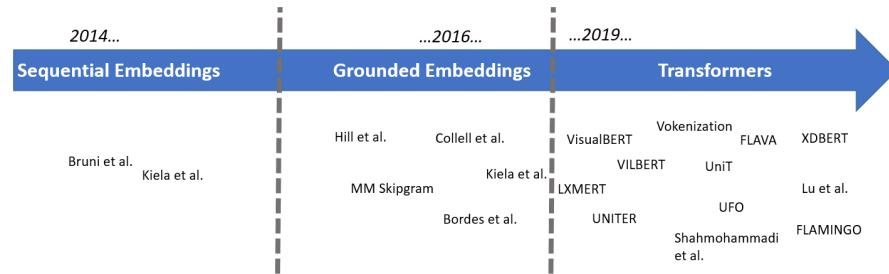


FIGURE 3.10: Historical evolution of models which integrate visual information into pure language models.

3.3.2 Word-Embeddings: Survival-Kit

In other parts of this books, the most important NLP-models and the latest developments in the field are extensively described. In this section, some information will be provided, which might be helpful to understand some of the aspects discussed in this subchapter. As it may have been inferred in the introduction, the starting point is always a pure language model, namely a model which employs only textual inputs in order to generate word embeddings, which are representations of words in form of numerical vectors. The most widely used pure language models in the papers presented in this subchapter are the following three:

- **Skipgram** (Word2vec, Mikolov et al. (XXXX)), where given a target word,

the probability of the neighboring (surrounding) words in a pre-defined window has to be maximized. Training takes place either through a *hierarchical softmax* or through *negative sampling*, which involves maximizing the probability of words which are real neighbors and minimizing that of words which are not real neighbors (the “negative samples”)

- **GloVe** ([Pennington et al. \(XXXX\)](#)), which is based on words co-occurrence across the *entire* corpus, with the goal of minimizing the difference between the dot product of the embedding vectors of two words and the logarithm of the number of co-occurrences
- **BERT** ([Devlin et al. \(XXXXd\)](#)): two pre-training tasks to obtain word-embeddings:
 - Masked Language Modelling (MLM): given a sentence with [MASK]ed tokens, the goal is to predict these masked tokens
 - Next Sentence Prediction (NSP): given two sentences A and B, the goal is to predict if B follows from A

Two additional remarks to conclude this section. First, Skipgram and GloVe generate embeddings which are “*context-free*”: they do not take into account the context in which words occur. On the contrary, BERT is designed to represent words given the context (sentence) in which they occur: we can thus have different embeddings for the same word, depending on the context. Second, the inputs of these models are *tokens*: with the help of a *tokenizer*, which can be different for different models, the text is split in “chunks”, called *tokens* (and they are not necessarily single words).

3.3.3 The Beginning: Sequential Multimodal Embeddings

Supposing we add linguistic and visual feature representations related to a particular word, how could we fuse them? One intuitive idea would be to *concatenate* the textual and visual modalities. Let V_{text} be the textual (vectorial) representation of a word and let V_{img} be its visual (vectorial) representation, a fused representation F of a certain word w might take the following simplified form:

$$F = \gamma(V_{text}) \bigoplus (1 - \gamma)V_{img}$$

where γ is a tuning parameter which controls the relative contribution of both modalities to the final fused representation. [Bruni et al. \(XXXX\)](#) propose a model where the meaning of a target word is represented in the form of a semantic vector and all vectors are collected in a *text-based semantic matrix*; textual embeddings are computed based on (transformed) co-occurrence counts of words in a pre-defined window. The starting point to obtain an image-based representation of certain target word is a dataset of labeled images. For each

image associated to the target word (which means that the target word is to be found in the image's caption), low-level features called “local descriptors” - which incorporate geometric information of specific areas of a certain picture - are extracted and then these descriptors are assigned to clusters (*bags*) of “visual words”¹. Afterwards, for each target word, visual word occurrences are summed up together to obtain the occurrence counts related to the target word. These image-based semantic vectors are then transformed and collected in an *image-based semantic matrix*. The two matrices are then concatenated and projected into a common latent multimodal space with a singular value decomposition. Thanks to this process a *textual mixed matrix* and a *visual mixed matrix* are extracted and then combined together according to different fusion strategies to build the multimodal embeddings. In this first, relatively cumbersome (historically motivated) example, the vector representation of an image is obtained with non-trivial features engineering.

In recent years, the use of neural networks has made an “automatic feature selection” possible. This is what for example [Kiela and Bottou \(XXXX\)](#) propose, extracting visual features from the first seven layers of a convolutional neural network (proposed by [Krizhevsky et al. \(XXXX\)](#)) trained on 1.6 million images from the ImageNet database ([Deng et al. \(XXXX\)](#)), which produces scores for 1,512 object categories. The linguistic part of the model relies on the Skipgram model by [Mikolov et al. \(XXXX\)](#) and consists of 100-dimensional vector representations. The multimodal representation is again obtained by concatenation of both modalities.

Another notable example of concatenation/sequential combination of textual and visual modalities is the work of [Silberer and Lapata \(XXXb\)](#): textual and visual modalities are represented by separate vectors of textual and visual attributes. During training, these textual and visual inputs vectors are separately fed to denoising (unimodal) autoencoders, the training objective of which is the reconstruction of a certain corrupted input - e.g. through masking noise - from a latent representation. Their outputs are then jointly fed to a bimodal autoencoder to be mapped to a multimodal space, on which a softmax layer (classification layer) is added, which allows the architecture to be fine-tuned for different tasks.

3.3.4 The Grounded Space

The aforementioned models assume implicitly a one-to-one correspondence between text and images: a visual representation is extracted only from words which are associated to a concrete image. This is a limitation, for two partially overlapping reasons. One one hand, how can we depict words for which no image is available in our training set? Is it possible to *imagine* visual representa-

¹See for example [Bosch et al. \(XXXX\)](#) for more details on this technique, called “bag-of-visual-words”.

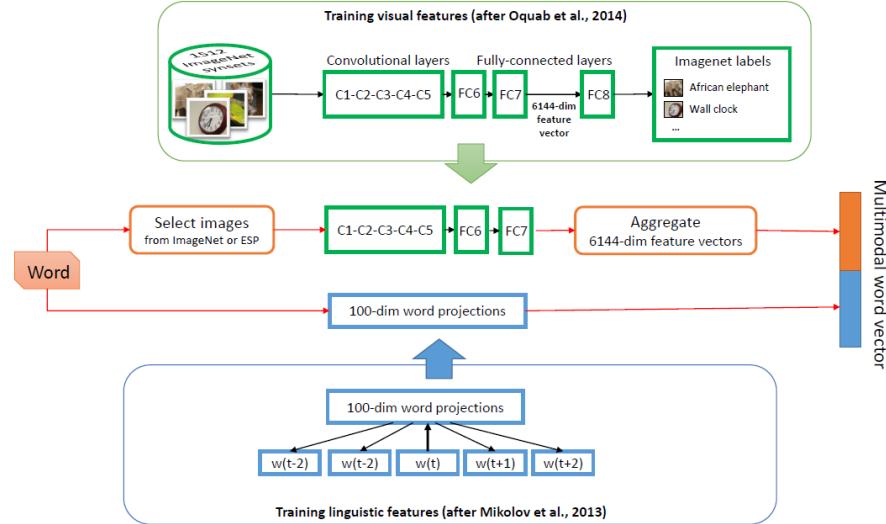


FIGURE 3.11: From @kiela2014learning. Textual and visual features vectors are concatenated.

tions purely from linguistic ones? On the other hand, could we hypothetically find a visual representation for each word? This might be true for concrete words but when it comes to abstract ones, it is not always possible to find suitable visual representations or, said in other terms, many words are not visually grounded. For this reasons, researches have addressed the question: could we map textual and visual elements to a grounded space and design models able to generalize images and words beyond those in the training set? Well, the answer is yes!

Lazaridou et al. (XXXX) propose a multimodal Skip-gram architecture where the objective function of a Skip-gram is “augmented” with an additional visual objective:

$$\frac{1}{T} \sum_{t=1}^T (\mathcal{L}_{ling}(w_t) + \mathcal{L}_{vision}(w_t))$$

where \mathcal{L}_{ling} is the Skip-gram loss function and \mathcal{L}_{vision} is the additional visual loss for the target word w_t . In particular, \mathcal{L}_{vision} has the form of a hinge loss, the goal of which is to make the (vectorial) linguistic representation of a certain word more similar to its visual representation:

$$\mathcal{L}_{vision}(w_t) = - \sum_{w' \sim P_n(w)} \left(\max(0, \gamma - \cos(z_{w_t}, v_{w_t}) + \cos(z_{w_t}, v_{w'})) \right)$$

where $v_{w'}$ is a visual representation of a randomly chosen word w' (drawn

from a probability distribution $P_n(w)$) used as negative sample, v_{w_t} is the corresponding visual vector and z_{w_t} is the target multimodal word representation which has to be learned by the model. It is nothing more than a linear transformation of a word representation u_{w_t} : $z_{w_t} = M^{u \rightarrow v} u_{w_t}$ and $M^{u \rightarrow v}$ is a cross-modal mapping matrix from linguistic inputs to a visual representation. It is important to remark that during training, for words which do not have associated images, \mathcal{L}_{vision} gets set to zero. When this cross-modal mapping matrix is estimated, it is then possible to find a visual representation for new words, which do not have a related image in the training set: the model allows to *imagine* new words. This is what is meant with grounded space: a perceptual (visual, in this case) space where a word is *grounded*, put in context.

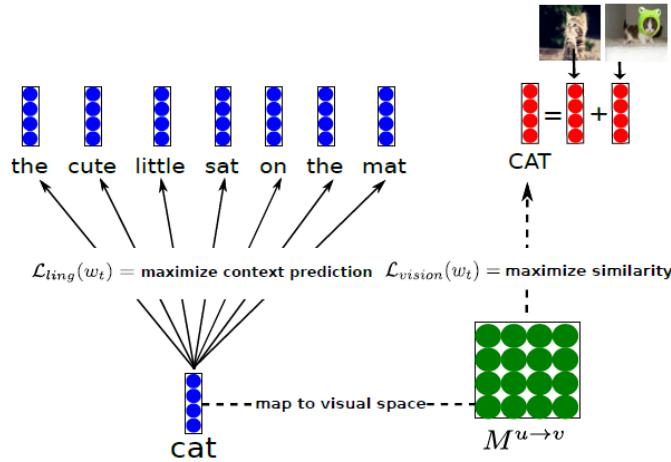


FIGURE 3.12: From @lazaridou2015combining. The linguistic embedding of the word ‘cat’ is mapped to a visual space, such that the similarity of vector representations of words and associated images is maximized.

Similar instances of a cross-modal mapping can be found for example in Kotturn et al. (XXXX) (a multimodal extension of the CBOW model specification of word2vec) and in Collell et al. (XXXX), where visual features are obtained from the forward pass of a CNN, pre-trained on ImageNet (Deng et al. (XXXX)) and a mapping function from the textual space to the visual space is obtained as a result of the training process. Also in this case it is possible to generate a visual representation from the embedding of a certain word, not necessarily present in the training set. In particular, they propose two specifications of the mapping function: a simple linear mapping and neural network with a single hidden layer. Last but not least, Hill and Korhonen (XXXX) recognize that concrete nouns are more likely to have a visual representation. For this reason, they map a set of concrete words (CSLB, Devereux et al. (XXXX)) to “bags of perceptual/visual features” and every time one of these

words is encountered during training, the Skip-gram model they are using stops training on that sentence and instead continues the training on a newly created “pseudo-sentence”, which takes into consideration the aforementioned bag of perceptual features. This list is unfortunately not exhaustive and there are other models with similar ideas, for example [Ailem et al. \(XXXX\)](#) or [Kiros et al. \(XXXX\)](#).

The aforementioned papers and related models focus on the modeling of semantics of words. Nonetheless, there are models designed to address tasks at sentence-level, such as sentiment analysis or sentence entailment. [Kiela et al. \(XXXX\)](#) employ a bidirectional Long Short-Term Memory (LSTM, [Hochreiter and Schmidhuber \(XXXX\)](#)) architecture to model sentence representations, in order to gain information from the text in both directions. The goal is again to encode a sentence and ground it in an image. Textual embeddings are obtained with GloVe ([Pennington et al. \(XXXX\)](#)) and they are then projected on a grounded space with a linear mapping. This grounded word vector serves as input for the bidirectional LSTM, which is trained together with the linear mapping. Their model is versatile and depending on the loss function specification, it can not only propose alternative captions to an image (which is a way to frame sentence equivalence tasks) but also predict captions from images or perform both tasks at the same time. This last point highlights an important characteristic of many of the models discussed in this subchapter: even though the focus is on the empowerment of pure language models with the addition of visual elements, some of the models discussed here can be used for purposes other than pure language tasks. The control over which task is performed is usually exercised by either specifying different loss functions (as in the last model described) or setting properly certain hyperparameters (such as in the previously described model by [Silberer and Lapata \(XXXXb\)](#)).

3.3.5 The Transformers Era

A turning point for the field of NLP was [Vaswani et al. \(XXXXb\)](#)’s paper “Attention is all you need”, where the authors proposed for two machine translation tasks a novel architecture, the Transformer (not to be confused with the giant robots from the Michael Bay’s blockbuster movies!), which leverages only the attention mechanism. Even though an exhaustive description of the Transformer architecture is beyond the scope of this subchapter, it is worth mentioning why they became so popular over the past four years in the field of NLP (among others), in comparison to Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs).

Well, the three main properties of Transformers are the following:

- Self-Attention
- Parallel input processing

- Positional embeddings²

When feeding for example a textual sentence to a RNN, the network deals with one word after the other in a sequential fashion and one of the known issues is the fact that information contained in earlier parts of the sequence tend to “fade away” as the sentence is analyzed further: newer inputs carry a larger influence on the outputs at a given step. LSTMs try to mitigate this problem by introducing a component called “gate”, which regulates the information flow, namely which information from the past inputs need to be “remembered” by the model. The goal is to capture long-term dependencies among different parts of the sentence fed into the model.

On the contrary, thanks to the Self-Attention mechanism, at each step Transformers can access previous steps, thus limiting to a minimum the loss of information. Moreover, inputs are processed not sequentially but all at the same time, thus allowing to capture dependencies by looking at the sentence *as a whole* and this could make a fundamental difference in many downstream applications: for example in the German language, in dependent clauses (“Nebensaetze”), the verb comes at the end of the phrase but it determines the verbal case of the nouns that come *before* the verb. Thus Transformer could potentially capture the dependencies between the verb coming at the end of the sentence and the words at the beginning. Lastly, Transformers encode for every input information on its position within a sentence, since it is often the case, that the importance and meaning of a certain word varies depending on its position within a sentence. These were the Transformers, in a nutshell.

But Transformers did not only bring a change of paradigm in terms of architectures. First, while for models in the pre-Transformers era described before, the focus was on the ability of word embeddings to capture similarity among words, now the focus has shifted more on downstream tasks (more on this later in the evaluation section), encompassing not only pure linguistic ones but also tasks with visual components, such as for example, visual question answering. It is now more difficult (but not impossible) to draw a line between models where “images support pure language models” (the object of this subchapter) and models which could be actually categorized as “vision and language” models but they can be employed also to solve pure linguistic tasks. This issue brings another peculiarity of many Transformers-base models, namely their “universal vocation”: without loss of generality we could say that the idea is now to design powerful (multimodal) pre-training (mostly *self-supervised*) tasks capable of generating task-agnostic representations, whose encoded knowledge can be efficaciously transferred to diverse downstream tasks, limiting the amount of labeled data necessary to fine-tune the models (this is the so-called *few-shot learning*).

Let’s briefly discuss two examples, Flava ([Singh et al. \(XXXX\)](#)) and UniT ([Hu](#)

²It may be argued that this point is a necessity to be able to work on sequences rather than a strength.

and Singh (XXXXa)). Flava has two separate encoders for images and text and a multimodal encoder, all based on the Vision Transformer (Dosovitskiy et al. (XXXX)). Unimodal pre-training consists of masked image modeling (where a set of image patches are to be reconstructed from other unmasked image patches) and masked language modeling. Multimodal pre-training tasks consist instead of a global contrastive loss (maximization of cosine similarities between paired images and text), a masked multimodal modeling (where image patches and text tokens are masked) and an image-text matching task. The model is pre-trained jointly on unimodal and multimodal datasets and then evaluated (fine-tuned) on 22 vision tasks, 8 pure linguistic tasks and 5 vision and language tasks.

UniT has an image encoder and a text encoder, a multimodal domain-agnostic decoder and task-specific heads. There is no pre-training on multimodal data and the model is trained end-to-end on 7 tasks (vision, language and vision an language) and 8 datasets, with the idea that solving different tasks across domains in a jointly fashion should prevent general knowledge from being lost due to fine-tuning over particular downstream tasks.

These two examples clearly show what it is meant by “universal vocation” of many modern Transformer-based models. But there are still models specifically designed to solve pure language tasks and in the following pages, two of them will be described.

3.3.5.1 Vokenization

It is often difficult for a child to describe the meaning of a certain word. A child might not be able to describe what a lion is but if he is given pictures of different animals he might be very well able to point at the picture of a lion. *Visual pointing* could thus act as a form of supervision to natural language. Is it possible to build within a pure language model a form of visual supervision, which mimics the visual pointing often adopted by children? This is exactly the problem that Tan and Bansal (XXXX) try to address: how to associate to each textual representation (token) a visual representation (Voken).

Let’s suppose we had a dataset of word(token)-image pairs. We could integrate in the pre-training framework of pure language models the following *Voken-Classification* task:

$$\begin{aligned} \mathcal{L}_{VOKEN-CLS}(s) &= -\sum_{i=1}^l \log p_i(v(w_i; s) | s) \\ \mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_l &= \text{language model}(w_1, w_2, \dots, w_l) \\ p_i(v|s) &= \text{softmax}_v \{W\mathbf{h}_i + b\} \end{aligned}$$

where $\{h_i\}$ is the feature representation of each token in a sentence $s = \{w_i\}$ extracted from a language model (such as BERT) and the vokens originate from a **finite** set of images X . Each h_i is then transformed into a probability

distribution through a softmax layer, with the voken-classification loss defined as the negative log-likelihood of all related vokens.

The model architecture would then be:

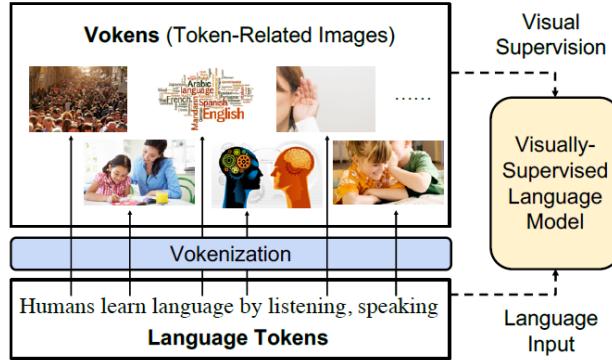


FIGURE 3.13: From @tan2020vokenization. Visually supervised the language model with token-related images, called Vokens.

Everything sounds fantastic! There is only one small pitfall: a set of X of images for all tokens does not exist! Could we find a proxy for such a set? One might consider image-captioning datasets such as MS COCO (Lin et al. (XXXb)). But also this suboptimal solution is problematic.

Dataset	# of Tokens	# of Sents	Vocab. Size	Tokens #/ Sent.	1-Gram JSD	2-Gram JSD	Grounding Ratio
MS COCO	7.0M	0.6M	9K	11.8	0.15	0.27	54.8%
VG	29.2M	5.3M	13K	5.5	0.16	0.28	57.6%
CC	29.9M	2.8M	17K	10.7	0.09	0.20	41.7%
Wiki103	111M	4.2M	29K	26.5	0.01	0.05	26.6%
Eng Wiki	2889M	120M	29K	24.1	0.00	0.00	27.7%
CNN/DM	294M	10.9M	28K	26.9	0.04	0.10	28.3%

FIGURE 3.14: From @tan2020vokenization. Statistics of image-captioning dataset and other natural language corpora. VG, CC, Eng Wiki, and CNN/DM denote Visual Genome, Conceptual Captions, English Wikipedia, and CNN/Daily Mail, respectively. JSD represents Jensen–Shannon divergence to the English Wikipedia corpus.

The *Grounding Ratio* is defined as the proportion of tokens in a dataset which are related to a specific visual representation (i.e. the tokens are *visually grounded*), such as “dog”, “table” and the like. In figure 3.14 it is striking that only around one third of tokens contained in pure language corpora such Wiki103, English Wikipedia and CNN/DM are visually grounded in im-

age captioning datasets³. It is not possible to rely (only) on image captioning datasets to build the Voken-Classification task. But the fact that a word/token does not have a visual representation in one of these datasets, it does not mean that it is not possible to visually represent the word/token. Would it be possible to associate images to words/tokens not directly visually grounded? Well, the answer is yes!

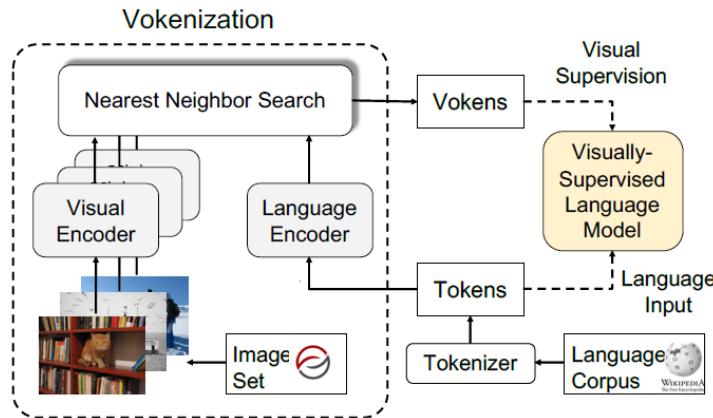


FIGURE 3.15: From @tan2020vokenization. The Vokenization process. A contextualized image (visual token, Voken) is retrieved for every token in a sentence and with this visual token, visual supervision is performed.

The **Vokenization** is a process to *assign* every token w_i contained in a sentence s to a visual representation (called *voken*) originating not from a generative model but rather from a finite set of images $X = \{x_1, \dots, x_n\}$. The voken $v(w_i; s)$ is the image from X which maximizes the following *Relevance Score Function*:

$$v(w_i; s) = \arg \max_{x \in X} r_{\theta^*}(w_i, x, s)$$

This function takes into account not only the token w_i itself, but also the context (the sentence) and it is parametrized by θ with θ^* being the optimal value (which has to be estimated).

3.3.5.1.1 The Relevance Score Function: Model, Training, Inference

The Relevance Score Function is defined as the inner product of the language feature representation $f_\theta(w_i, s)$ and the visual feature representation $g_\theta(x)$:

Supposing h_1, \dots, h_l and e are the embeddings originating from pre-trained language and visual encoders respectively (in the paper the authors use BERT

³From an operative point of view, the authors consider a token type “visually grounded” if it has more than 100 occurrences in MS COCO

and ResNeXt), the language and visual representations are obtained first by applying multi-layer perceptrons w_mlp_θ and x_mlp_θ to downproject the embeddings from the pre-trained models to a common vector space and secondly they are normalized (with L2-Norm):

$$\begin{aligned}\mathbf{f}_\theta(w_i; s) &= \frac{w_mlp_\theta(\mathbf{h}_i)}{\|w_mlp_\theta(\mathbf{h}_i)\|} \\ \mathbf{g}_\theta(x) &= \frac{x_mlp_\theta(\mathbf{e})}{\|x_mlp_\theta(\mathbf{e})\|}\end{aligned}$$

With respect to the training of the model, to estimate the optimal value for the parameter θ , image-captioning datasets, which are collections of sentence-image pairs, are employed. Operationally, for every sentence s_k associated to image x_k in the image-captioning dataset, each token w_i in s is associated to x_k and the *hinge loss* is used to estimate the optimal value of θ^* :

$$\mathcal{L}_\theta(s, x, x') = \sum_{i=1}^l \max(0, M - r_\theta(w_i, x, s) + r_\theta(w_i, x', s))$$

The goal is to maximize the Relevance Score Function between aligned token-image pairs $(w_i, x; s)$ and to minimize the score for unaligned pairs $(w_i, x'; s)$ by at least a margin M , with x' being a randomly sampled image from the image captioning dataset **not** associated to sentence s .

Once we have the language feature representation $f_\theta(w_i, s)$ for each token in our language corpus and the optimal estimate of θ , how is it possible to find the image x encoded with the visual feature representation $g_\theta(x)$, which maximizes the Relevance Score Function? As said earlier, the function is expressed as the inner product of the textual and visual representations and since the feature vectors have euclidean norm equal to 1, the inner product maximization problem is equivalent to a nearest neighbor search problem. It is just sufficient to find the vector $g_\theta(x)$ which is the nearest neighbor of $f_\theta(w_i, s)$ ⁴.

⁴The proof is straightforward. Let $X \in \mathbb{R}^l$ and have euclidean norm equal to 1, which means $\|X\|_2 = 1$. In the nearest neighbor search we need to find the vector $Y \in \mathbb{R}^l$, also with norm equal to 1, which has minimal euclidean distance with X . This is the quantity to be minimized:

$$\begin{aligned}d(X, Y) &= \sqrt{\sum_{i=1}^l (x_i - y_i)^2} \\ &\stackrel{\text{squared}}{=} \sum_{i=1}^l x_i^2 + \sum_{i=1}^l y_i^2 - 2 \sum_{i=1}^l x_i y_i \\ &= \|X\|_2^2 + \|Y\|_2^2 - 2X^T Y \\ &\stackrel{\text{Norm}=1}{=} 1 + 1 - 2X^T Y \\ &= 2(1 - X^T Y)\end{aligned}$$

And through these simple algebraic manipulations, it is possible to see that minimizing

With this process, it is thus possible to assign a visual representation, a voken, to any word/token in a language corpus, pooling from a finite set of images. The problem of the low Grounding Ratio outlined above is solved and the Voken-Classification task could be integrated in the pre-training framework of any pure language model. Moreover, the authors propose a method called *Revokenization*, which allows to transfer vokens generated using a particular tokenizer to frameworks which employ other tokenizers.

3.3.5.2 One Step Further: The Power Of Imagination

Wikipedia defines *imagination* as “the production or simulation of novel objects, sensations, and ideas in the mind without any immediate input of the senses”. Indeed, humans do not only associate words with real images, but also leverage the ability to *imagine* words/concepts: imagination can help the human brain solve problems with limited supervision or sample points by empowering its generalization capabilities. Until now we discussed language models supported by visual information in form of *real* images (e.g. those retrieved from image-captioning datasets). But with the recent advancements in the field of generative models for images, it is for sure worth investigating if these generative models can help pure language models to produce better representations of words. In particular, the framework proposed by **Lu et al. (XXXXc)**, **iACE (Imagination-Augmented Cross-Modal Encoder)** will now be discussed: the idea is simply to use a generative model to obtain a visual representation of a textual input and then use these imagined representations as “imagination supervision” to pure language models.

This framework has two main components:

- the **imagination generator** G : given an input text x , VQGAN (Esser et al. (XXXXb)) is used to render an “imagination” i of x and CLIP (Radford et al. (XXXXa)) is used to see how well the generated image i is aligned to the input text x . This generative framework is known as VQGAN+CLIP
- **Cross-modal Encoder** E_c : the input text and the rendered imagination are firstly encoded with a language and a visual encoder respectively and then CLIP is employed as cross-modal encoder with inputs being text-imagination pairs

The learning procedure is composed of two main steps (depicted in figure 3.16): the first step consists in the pre-training of a visually supervised Transformer. In particular, the Voken-Classification task described before is employed, alongside a masked language modeling task. This is the baseline model,

the euclidean distance between X and Y is equivalent to maximize $X^T Y$, which is the inner product. This proves the equivalence between inner product maximization and nearest neighbor search.

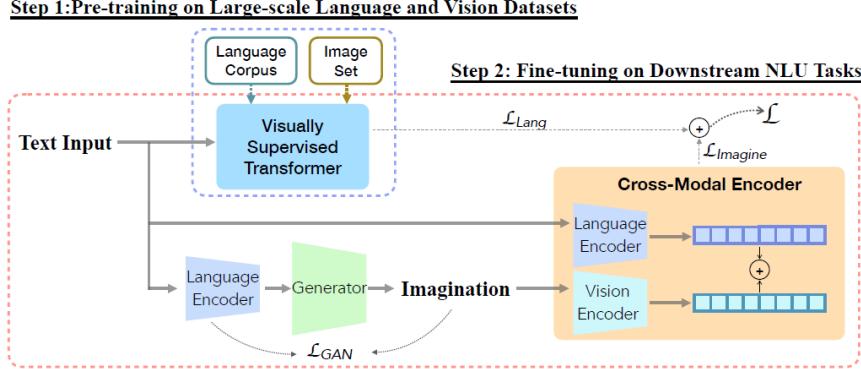


FIGURE 3.16: From @lu2022imagination. The generator G visualize imaginations close to the encoded texts by minimizing \mathcal{L}_{GAN} . The cross-modal encoder E_c learns imagination-augmented language representation. Two-step learning procedure consists of: 1) pre-train a Transformer with visual supervision from large-scale language corpus and image set, 2) fine-tune the visually supervised pre-trained Transformer and the imagination-augmented cross-modal encoder on downstream tasks.

where no information from the “imagination” procedure comes yet into play. The second step is the *imagination-augmented fine-tuning* with two downstream datasets D (GLUE, Wang et al. (XXXXa) and SWAG, Zellers et al. (XXXXb)).

On one side, the visually-supervised Transformer (the baseline) relies only on the textual input during the fine-tuning phase and the following loss function is employed:

$$\mathcal{L}_{Lang} = - \sum_{j=1}^{|D|} \sum_{k=1}^K y_k \log p_k(d_j(t)|D)$$

On the other hand, the *iACE* is trained to minimize the following cross-entropy loss:

$$\mathcal{L}_{Imagine} = - \sum_{j=1}^{|D|} \sum_{k=1}^K y_k \log p_k(d_j(t, v)|D)$$

with t and v being the textual and imagined features representations respectively, j indicates the j -th data sample in dataset belonging to dataset D , K is the number of classes and p_k is the conditional distribution of d_j . Training takes place in a jointly fashion and both losses, the imagination-augmented one $\mathcal{L}_{Imagine}$ and the pure language loss \mathcal{L}_{Lang} are linearly combined, with λ being a balance factor:

$$\mathcal{L} = \lambda \mathcal{L}_{Imagine} + (1 - \lambda) \mathcal{L}_{Lang}$$

To sum up, this model-agnostic framework uses *generated images* for visual supervision and could be integrated on top of pure language models (such as BERT) or visually supervised models (such as the Voken model, which uses Vokens, real images for visual supervision).

3.3.6 Was It Worth?

In this subchapter we investigated how visual inputs can support pure language models in capturing the semantics of words. We started with simple concatenation of linguistic and visual features and ended up with Transformer-based models, which are able to shape different word embeddings for the same word by taking into account also the context (the sentence). But now the question arises: with the addition of visual information, do we obtain word embeddings that are better than those from pure language models? In other words, is what we all have so far discussed worth? Well, as it is often the case in scientific research, the answer is: “it depends!”

Individual evaluation of each single model might not be ideal because each model has its peculiarities and it is impractical to make a direct comparison among them. It is more useful to capture and discuss the themes which are common to many models, in order to understand their strengths and weaknesses. This is how we will proceed and we will also differentiate between evaluation before Transformers and evaluation after Transformers.

3.3.6.1 Evaluation In The Pre-Transformers Era

Before the advent of Transformers, the evaluation focus was on the degree of alignment between learned semantic representations (word embeddings) and representations by human speakers, in form of correlation between model-based and human-based word-similarity judgments. Three main types of similarity are usually considered:

- Semantic similarity, e.g. “pasta is similar to rice”
- Semantic relatedness, e.g. “Bear is related to mountain”
- Visual similarity, e.g. “cucumbers look like zucchinis”

The evaluation pipeline could be summarized as follows:

Word embeddings are vectors and to measure the degree of similarity between two vectors, the *Cosine Similarity* is often used in the literature. In an ideal

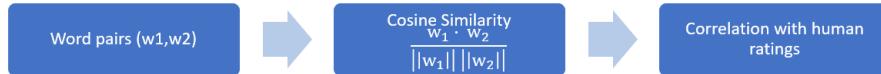


FIGURE 3.17: Pipeline for intrinsic evaluation of semantic representations. In the first step, the cosine similarity between two word embeddings w_1 and w_2 is used as similarity measure and in a second step, the correlation with human speakers' assessment is computed to gauge the quality of the embeddings. The higher the correlation, the better the embeddings.

setting, we would have word embeddings with the following characteristics: if two words are semantically similar, the two embedding vectors should be similar and their cosine similarity should go towards 1. If the two words are unrelated, the embedding vectors should be orthogonal to each other and as a consequence, the cosine similarity should go towards zero. Lastly, if two words are negatively related, the two embedding vectors should point at opposite directions and the cosine similarity should go towards -1. Once these similarity measures between word pairs are computed, in order to measure the quality of the embeddings several benchmarks can be employed, such as MEN (Bruni et al. (XXXX)), WordSim353 (Agirre et al. (XXXX)) and SimLex999 (Hill et al. (XXXX)). These datasets could be described as collections of word pairs and associated similarity ratings by human speakers. Operationally, this means that real people were asked if a pair of words was related or not and to which degree, on a scale between -1 (negatively related) to +1 (semantically equivalent). The higher the correlation between the cosine similarity and the similarity judgments by humans, the higher the quality of the word embeddings. Having done this methodological premise, let's discuss the performance of these pre-Transformer models!

Since the goal of these models is to enhance pure language models with the addition of visual inputs, the baseline in the evaluation is always one (or more) pure language model(s). Well, do visually grounded embeddings outperform non-grounded ones? What emerges from virtually all papers is that visual grounding can actually help get a better semantic representation of *concrete* concepts, such as "cat", "table", "bicycle", whereas they do not help much with the representation of abstract concepts such as "love" and "peace".

In figure 3.18 we can see that pure language models still perform better than models with visual inputs when it comes to the representation of abstract nouns. Another example is Kiel et al. (XXXX): they found that their models perform better when tested on datasets with a higher degree of concreteness and the same conclusion is reached by Collell et al. (XXXX), which state that visual information can empower the representations of concepts that are to a certain extent visual. To sum up, effective semantic representation of abstract concepts constitute the main limitation common to many of the models discussed in this section.

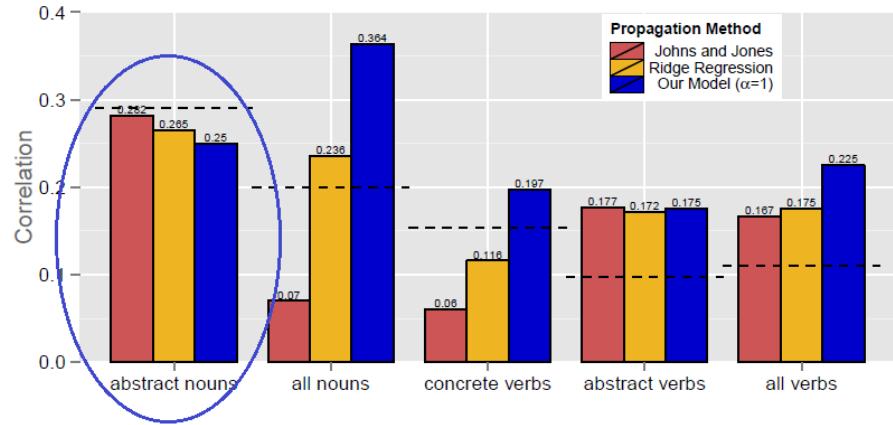


FIGURE 3.18: From @hill2014learning: Each bar represents a different model settings and the dashed line indicates the pure linguistic benchmark model.

3.3.6.2 Evaluation In The Post-Transformers Era

A limitation of the *intrinsic* evaluation metrics is the high degree of subjectivity: the *similarity* between two concepts depends in many instances on the experience, cultural background and preferences of the human observers. This is why the evaluation focus has now shifted to a more *extrinsic* dimension: how well do the models perform in downstream tasks? The problem of the “lack of objectivity” is thus solved because on downstream tasks there is no room for opinions. The datasets used to train the models are also different and the most widely used are:

- GLUE (Wang et al. (XXXXa)): 9 tasks, including single-sentence tasks (e.g. sentiment analysis), similarity tasks (e.g. paraphrasing), inference tasks (e.g. textual entailment)
- SQuAD (Rajpurkar et al. (XXXXb)): question/answer pairs
- SWAG (Zellers et al. (XXXXb)): multiple choice questions about grounded situations

As previously discussed, many Transformer-based models have universal vocation: they are built to solve a heterogeneous range of tasks from the language and vision domain. If we thus consider only performance on pure language tasks, the following two tables from Tan and Bansal (XXXX) are insightful:

It is straightforward: unlike in the pre-Transformers Era, where grounded word embeddings could improve performance over baselines, Transformer-based universal models **do not** outperform pure language models such as BERT or RoBERTa. Nonetheless, the addition of visual supervision (the

Model	Init. with BERT?	Diff. to BERT Weight	SST-2	QNLI	QQP	MNLI
ViLBERT (Lu et al., 2019)	Yes	0.0e-3	90.3	89.6	88.4	82.4
VL-BERT (Su et al., 2020)	Yes	6.4e-3	90.1	89.5	88.6	82.9
VisualBERT (Li et al., 2019)	Yes	6.5e-3	90.3	88.9	88.4	82.4
Oscar (Li et al., 2020a)	Yes	41.6e-3	87.3	50.5	86.6	77.3
LXMERT (Tan and Bansal, 2019)	No	42.0e-3	82.4	50.5	79.8	31.8
BERT _{BASE} (Devlin et al., 2019)	-	0.0e-3	90.3	89.6	88.4	82.4
BERT _{BASE} + Weight Noise	-	6.5e-3	89.9	89.9	88.4	82.3

FIGURE 3.19: From @tan2020vokenization. Results of vision-and-language pre-trained models (universal models) on GLUE tasks compared to baseline models (BERT).

Method	SST-2	QNLI	QQP	MNLI	SQuAD v1.1	SQuAD v2.0	SWAG	Avg.
BERT _{6L/512H}	88.0	85.2	87.1	77.9	71.3/80.2	57.2/60.8	56.2	75.6
BERT _{6L/512H} + Voken-cls	89.7	85.0	87.3	78.6	71.5/80.2	61.3/64.6	58.2	76.8
BERT _{12L/768H}	89.3	87.9	83.2	79.4	77.0/85.3	67.7/71.1	65.7	79.4
BERT _{12L/768H} + Voken-cls	92.2	88.6	88.6	82.6	78.8/86.7	68.1/71.2	70.6	82.1
RoBERTa _{6L/512H}	87.8	82.4	85.2	73.1	50.9/61.9	49.6/52.7	55.1	70.2
RoBERTa _{6L/512H} + Voken-cls	87.8	85.1	85.3	76.5	55.0/66.4	50.9/54.1	60.0	72.6
RoBERTa _{12L/768H}	89.2	87.5	86.2	79.0	70.2/79.9	59.2/63.1	65.2	77.6
RoBERTa _{12L/768H} + Voken-cls	90.5	89.2	87.8	81.0	73.0/82.5	65.9/69.3	70.4	80.6

FIGURE 3.20: From @tan2020vokenization. Fine-tuning results of different pre-trained models w/ or w/o the voken classification task (denoted as “Voken-cls”).

Voken-Classification task) in the pre-training framework can boost performance above the level of pure language models.

Pezzelle et al. (XXXX) analyzed the *intrinsic* quality of embeddings of some vision and language (“universal”) models:

From this *intrinsic* evaluation perspective (which was popular in the pre-Transformers Era), vision and language models do not generally outperform domain-specific models such as BERT and also in this case the only real competitor of pure language models is a model with visual supervision (again, Vokenization).

The bar plots depict correlation between human- and model-based similarity ratings, differentiating between the most *concrete* concepts contained in a certain dataset⁵ and the whole dataset (thus including more abstract concepts). The results confirm the trend: multimodal models are more effective than pure

⁵See Brysbaert et al. (XXXX) for information on how *concreteness* of a word can be estimated.

model	input	Spearman ρ correlation (layer)			
		RG65	WS353	SL999	MEN
BERT-1M-Wiki*	L	0.7242 (1)	0.7048 (1)	0.5134 (3)	–
BERT-Wiki <i>ours</i>	L	0.8107 (1)	0.7262 (1)	0.5213 (0)	0.7176 (2)
GloVe	L	0.7693	0.6097	0.3884	0.7296
BERT	L	0.8124 (2)	0.7096 (1)	0.5191 (0)	0.7368 (2)
LXMERT	LV	0.7821 (27)	0.6000 (27)	0.4438 (21)	0.7417 (33)
UNITER	LV	0.7679 (18)	<u>0.6813 (2)</u>	<u>0.4843 (2)</u>	0.7483 (20)
ViLBERT	LV	<u>0.7927 (20)</u>	0.6204 (14)	0.4729 (16)	<u>0.7714 (26)</u>
VisualBERT	LV	0.7592 (2)	0.6778 (2)	0.4797 (4)	0.7512 (20)
Vokenization	LV	0.8456 (9)	0.6818 (3)	0.4881 (9)	0.8068 (10)

FIGURE 3.21: From @pezzelle2021word. Spearman’s rank correlation between similarities computed with representations by all tested models and human similarity judgments in the five evaluation benchmarks.

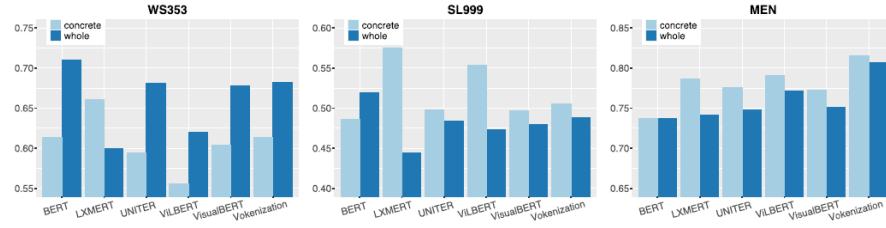


FIGURE 3.22: From @pezzelle2021word. Correlation between model and human similarity ratings on WordSim353, SimLex999 and MEN. Each barplot reports results on both the whole benchmark and the most concrete subset of it.

language models at representing concrete words but in many instances they still lag behind when it comes to more abstract concepts.

Last but not least, few words need to be spent on a topic which has been steadily gaining relevance: **Few-Shot Learning**. To train and test models, a large pool of paired images and texts is often needed and the creation of many of the datasets used in fine-tuning required a huge data collection effort, which had to be performed by human agents. This implies that the creation of such data pools can be very costly. For this reason, there is a growing interest in creating models able to cope with low-resource settings. This boils down to the question: can a model perform well on downstream tasks even with just a *limited number* of training examples? The goal is actually once again, to mimic how humans learn: a person does not need to see one thousand pictures of a table, to be able to recognize a table...

This table from Lu et al. (XXXXc), where models are trained using only up to 5% of the training set, shows for example the ability for a model supervised

	SST-2			QNLI			QQP			MNLI		
	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%
Extreme Few-shot												
<i>VOKEN(Bert_{base})</i>	54.70	77.98	80.73	50.54	51.60	61.96	44.10	60.65	65.46	37.31	54.62	58.79
<i>iACE(Bert_{base})</i>	77.98	80.96	81.42	51.64	58.33	64.03	49.36	63.67	71.17	40.07	56.49	59.57
<i>VOKEN(Roberta_{base})</i>	70.99	71.10	77.86	54.37	62.23	65.78	62.32	67.25	70.18	48.59	49.76	58.23
<i>iACE(Roberta_{base})</i>	75.34	78.66	83.60	54.79	65.03	65.83	65.43	68.11	70.77	48.94	52.74	59.39
Normal Few-shot	1%	3%	5%	1%	3%	5%	1%	3%	5%	1%	3%	5%
<i>VOKEN(Bert_{base})</i>	81.40	86.01	84.75	64.17	77.36	80.19	72.55	78.37	80.50	60.45	62.73	72.35
<i>iACE(Bert_{base})</i>	82.45	87.04	86.47	65.09	79.54	80.52	74.31	78.69	80.52	62.15	70.43	73.73
<i>VOKEN(Roberta_{base})</i>	83.78	84.08	87.61	75.00	81.16	81.23	73.14	79.09	79.63	63.51	70.68	74.02
<i>iACE(Roberta_{base})</i>	83.83	84.63	89.11	79.35	81.41	81.65	73.72	79.38	79.81	65.66	70.76	74.10

FIGURE 3.23: From @lu2022imagination. Model-agnostic improvement in Few-shot Setting with GLUE benchmark.

with “imagination” (which was a generated visual representation of a certain textual input) to outperform models with only simple visual supervision (the Voken-model). This is just an example, but the ability to perform well in *few-shot* settings has become the touchstone of the evaluation modern multimodal models.

3.3.7 The End Of This Story

We started this story with the *Symbol Grounding Problem*, which affirms that to grasp the meaning of a word, the word has to be put in a context other than the pure linguistic one. We thus investigated some of the architectures proposed to ground words in a visual space in form of static images. The goal (hope) is to better capture the semantics of words, in form of better word embeddings, to be employed in heterogeneous tasks, from *semantic-similarity* to downstream tasks, such as *sentiment analysis*.

From this brief analysis it emerges that grounding words in images can actually improve the representation of *concrete* concepts, whereas visual grounding does not seem to add value to pure language models when it comes to *abstract* concepts. Nonetheless, forms of visual supervision like the *Voken-Classification* task or the employment of generative models which allow to *imagine* words, such as in the *iACE-Framework*, might be the right way to bridge this gap. The Transformers have been a revolution in the field of NLP and with their advent, the trend has now become to build models with pre-training tasks capable of generating powerful task-agnostic word representations. The knowledge gained with these tasks can be then transferred to downstream tasks with the goal to limit the amount of labeled data necessary to fine-tune models. Labeling data is indeed costly: this is why the ability of a model to generalize well when exposed to just few training examples has been steadily gaining importance as evaluation metric. This was the so called *few-shot learning*. Moreover, Transformer-based models have “universal vocation”: they tend to

be multimodal and multi-task, encompassing vision, language and vision and language tasks. This idea might be appealing because humans learn by being exposed to a multitude of different inputs and tasks. But as we have seen, pure language models such as BERT tend to still outperform multimodal multi-task models. There is definitely room for improvement.

One might wonder whether the grounding of words in images is the right way to seek a better representation of words. Well, humans learn using all five senses and maybe the answer might be to incorporate in the models more heterogeneous perceptual information: not only static images but also videos, speech and the like. The debate is still open: the story *goes on...*

Last but not least, a mention needs to be made on concrete applications of these image-empowered word-embeddings. The use of images to support linguistic models has been experimented in several fields, from *Dialogue Response Generation* (e.g. Sun et al. (XXXX)) to *Machine Translation*, where for example Ive et al. (XXXX) found images to improve the quality of translation when the textual context is generic and/or ambiguous. The number of potential applications of the models described in this subchapter is growing steadily in the scientific community. But this is yet *another* story...

3.3.8 Appendix: Selected Models - Summary

The following table contains a summary of selected language models augmented with visual components. For each model, the following information are reported:

- Pure language model and pretraining data
- Visual features and pretraining data
- Fusion strategy of the two modalities
- Benchmarks/baselines for evaluation:
 - better performance over baseline(s)
 - mixed performance results over baseline(s)
 - worse performance over baseline(s)

The table is available in a more readable format [here](#).

Year	Author(s)	Page	Language	Visual Model	Text Model	Multimodal Model	Testset	Baseline(s)/model settings/comparison to
2018	Yeh et al.	1	English	Image	Text	Multimodal	Ether	
2018	Elia et al.	1	English	Image	Text	Multimodal	WordSim-353	Word
2018	Nam et al.	1	English	Image	Text	Multimodal	MEN	mixed representations
2018	Bleket et al.	1	English	Image	Text	Multimodal	BERT	bed-enhancing
2018	Khanh et al.	1	English	Image	Text	Multimodal	MRPC	embeddings
2018	Tran et al.	1	English	Image	Text	Multimodal	MRPC	only-purely
2018	Wado et al.	1	English	Image	Text	Multimodal	MRPC	Vi-sual
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	mixed embeddings
2018	Batoni et al.	1	English	Image	Text	Multimodal	MRPC	• No alternative
2018	“Multi-modal”	1	English	Image	Text	Multimodal	MRPC	embeddings
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	• Only used as a means of comparison.
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	ver-sions
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	of
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	fea-ture
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	and
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	scor-ing
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	level
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	fu-sion
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	model
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	settings-
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	Several
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	“fine tuned”
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	ver-sions
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	of
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	fu-sion
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	and
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	scor-ing
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	level
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	fu-sion
2018	Marco et al.	1	English	Image	Text	Multimodal	MRPC	model

Year	Language Model	Visual Model	Training Data	Presentation	Testset	Baseline(s)/model set
2011	Hall, Fe- lix, and Anna Ko- rho- nen. “Learn- ing ab- stract con- cept em- bed- dings from multi- modal data: Since you prob- ably can’t see what I mean.” Pro- ceed- ings of the 2014 Con- fer- ence on Em- pirical Methods in Natural Lan-	Skip ¹⁰⁰ Mapping Text8words Corpuso a bag of per- cep- tual fea- tures b(w), ex- tracted from ex- ter- nal sources and en- coded in an asso- cia- tive array P. Gen- era- tion of pseudo sen- tences based on these per- cep- tual fea- tures to be fed into the lan-	IMG- Pre- training sourc- (LM)	Multimodal training presentation and model description	ether	tings/comparison to Testset/Ether tuning modelResults

Extension of the ESP- Skipgram injecting Game perceptual (100K) information by generating pseudo-sentences CSLB based on a bag-of-Prop- visual-words. A hyperparameter α Norms controls the level of perceptual information relative to linguistic input.

Dataset. Concepts, which can directly be represented in the perceptual modality (e.g. concrete verbs and nouns) Propagation of perceptual features from canonical concepts (nouns and verbs) to enhance the representation of abstract verbs, those for which no direct modalities in the visual SVD space is available Abstract nouns (for which is more difficult nated to find a concrete visual modal representations. are still more efficiently learned

Year	Language	Model	Training	Pre-training	Multimodal training	Presentation	and source	Model description	Testset	Baseline(s)/model set	Comparison to	Model tuning	Results
2014	English	Skipgram	Seventh-layer	Concatenation of visual and textual embeddings.	-	-	CNN-MEN	Skip-gram	Mean	Word-gram	better on Sim353	(text- MEN: it only averaging cap- baselin)g	CNN-ture capture
Kiela et al.	Text	Text	Corpora	a-	geNet	-	-	not	Em- related-	Word- gram	better on Sim353	(text- MEN: it only averaging cap- baselin)g	Mean
Léon et al.	Image	CNN	(400M words)	(12.5M images)	-	-	-	only	bed- ness	-	-	-	Mean
Botou et al.	Image	Text	tract	Esp-Game	-	-	-	"re- dings"	dings better.	-	-	-	Max
2014	Image	Text	Naive	fida-	(100K images)	-	-	lated-	CNN- vi-	-	-	-	CNN- Max
Learn-	Image	Text	Im-	Cortures	images).	-	-	ness"	sual	but	also	only.	WordSim353.
ing	Image	Text	age	pus vec-	(100M words)	-	-	"similarity".					
Im-	Image	Text	Em-	pus vec-	(100M words)	-	-						
age	Image	Text	bed-	bed-	bed-	-	-						
Em-	Image	Text	dings	words)	us-	-	-						
us-	Image	Text	us-	im-	ing	ages,	-						
ing	Image	Text	vo-	ob-	Con-	ob-	-						
Con-	Image	Text	lu-	tained	vo-	tained	-						
lu-	Image	Text	tional	in	lu-	in	-						
tional	Image	Text	Neu-	two	tional	two	-						
Neu-	Image	Text	ral	ways:-	Neu-	ways:-	-						
ral	Image	Text	Net-	CNN-	ral	CNN-	-						
Net-	Image	Text	works	Mean	Net-	Mean	-						
works	Image	Text	for	(av-	works	(av-	-						
for	Image	Text	Im-	erage	for	erage	-						
Im-	Image	Text	proved	of all	Im-	of all	-						
proved	Image	Text	Multi-	fea-	proved	fea-	-						
Multi-	Image	Text	Modal	tures	Multi-	tures	-						
Modal	Image	Text	Se-	vec-	Modal	Se-	-						
Se-	Image	Text	man-	tors	Se-	vec-	-						
man-	Image	Text	tics.	rep-	man-	tors	-						
tics.	Image	Text	In	re-	tics.	rep-	-						
In	Image	Text	Pro-	sent-	In	re-	-						
Pro-	Image	Text	ceed-	ing	Pro-	sent-	-						
ceed-	Image	Text	ings	images)-	ceed-	ing	-						
ings	Image	Text	of	-	ings	images)-	-						
of	Image	Text	the	CNN-	of	-	-						
the	Image	Text	2014	Max	the	CNN-	-						
2014	Image	Text	Con-	(component-	2014	Max	-						
Con-	Image	Text	fer-	wise	Con-	(component-	-						
fer-	Image	Text	ence	max-	fer-	wise	-						
ence	Image	Text	on	i-	ence	max-	-						
on	Image	Text	Em-	mum	on	i-	-						
Em-	Image	Text	piri-	of all	Em-	mum	-						
piri-	Image	Text	cal	fea-	piri-	of all	-						
cal	Image	Text	Meth-	tures	cal	fea-	-						
Meth-	Image	Text	ods	vectors)	Meth-	tures	-						
ods	Image	Text	.	.	ods	vectors)	-						

LM-Visual Lang- mode YePape	Pagele- trainings Paper (LM)	IMG- Pre- training sourc (LM)	Multimodal training and sourc IMG)	Testset/ tuning model	Baseline(s)/model set- tings/comparison to ether modelResults
20\$ilbereVectors Ca- rina, Mirellaat- Lap- ata. “Learn ing grounded mean- ing rep- re- sen- ta- tions with au- toen- coders.” ceed- ings of the 52nd An- nual Meet- ing of the As- soci- a- tion for Com- pu- ta- tional Lin- guis- tics (Vol- ume 1: Long D	Vectors of McRae- tex- et sual as and tual al.sat- tributes are ata. are extracted. “Learn ing grounded mean- ing rep- re- sen- ta- tions with au- toen- coders.” ceed- ings of the 52nd An- nual Meet- ing of the As- soci- a- tion for Com- pu- ta- tional Lin- guis- tics (Vol- ume 1: Long D	Same Stacked dataset (denoising) as autoencoders for in each single butesSil- berer outputs are concatenated and al. (2013): to a stacked tax- bimodal on- autoencoder which omy map the inputs to of a joint hidden 636 layer. vi- su- alat- tributes (e.g., has wings, made of wood) and- nearly 700K im- ages from Im- a- geNet (Deng et al., 2009) de- scrib- ing more than 500 of McRae et al.’s (2005) nouns.	With - McRae Uni- et al.’s modal (2005), au- two to en- tasks:- coders word only- similarity- word Ker- categorizati ized Not Canonwidely ical used in Cor- the field. re- la- tion, Hardoon et al. (2004)- Bruni et al. (2014).		

Year	Paper	Language model	Visual model	Pre-training	Multimodal training	Presentation and source	Model description	Testset	Baseline(s)/model set	Comparison to	Model tuning	Results
2015	Lazaridou et al.	Skipgram	Word2Vec	ImageNet	The objective function is a linear composition of the language objective and the visual objective. The skipgram and a visual objective are proposed:- MM Skipgram A (MMSA): aligning vectors of visual and linguistic representations. Skipgram B (MMSB): estimate a cross-modal mapping matrix from linguistic onto visual representations.	-	-	Both	MEN- and VisSim.	Kiela et al. (2014) than simpler Bruni et al. (2014) / vision only, Sil- berer & La- p- ata (2014) related- ness and gram visual (text- similarity, baselines despite having Embed- bed- dings other visual models	MMSA and Bot- MMSB tou better than (linguis- al. (2014) / vision & SVD) MMSA	Baseline(s)/model set

LM-Visual Lang- mode Year	Pagele- eelingts Paper	IMG- Pre- trainings (LM)sourc (LM)	Multimodal training sourc model	presentation and model description	Testset/ ether tuning	Baseline(s)/model set- tings/comparison to modelResults
2017 Collell, Guillen, Ted Zhang, and Marie- Francine Moens. “Imag- ined vi- sual rep- re- sen- ta- tions as mul- ti- modal em- bed- dings.” Pro- ceed- ings of the AAAI Con- fer- ence on Ar- tifi- cial In- telli- gence. Vol. 31. No. 1. 2017.	300- Common Guillen, Craw- Ted GloVe- cor-tract pus,vi- 840Bual to- fea- kenstures, 2.2Mthe “Imag- ined hid- den layer of a CNN is taken. For each con- cept, two dif- fer- ent ways to com- bine the ex- tracted vi- sual features:- Aver- aging (av- erag- ing of al fea- tures vectors)-	Image Mapping from language to vision. No need of 1:1 correspondence between linguistic and visual inputs. Two different mappings are considered:- Linear (MAP-Clin)- Neural Network (MAP-Cnn).	- -	- - Outperformance Kiela in all WordSim350- Bot- where SemSim-tou (2014)have as- Simlex999- SimVerb3500- dou the VisSim. et al. (2015) Performance on the Sil- zero-shot berer learning & still La- inferior in p- many ata instances (2014)to the textual GloVe baselines. (text- only baseline)-	Concatenation.	

LM-Visual Lang mode Ye Paper	Page- le- train- Paper	IMG- Pre- train- (LM) sourc- (IMG)	IMG- Multimodal training and sourc- model	Baseline(s)/model set- tings/comparison to Testset/ ether tuning modelResults
2018 Krela, - Welch Douwe, et al. “Learned embedding vi- su- ally rep- re- sen- sen- ta- tions.” arXiv preprint arXiv:1707.06320 (2017).	Hongel GloVe ob- ained from the final ground- LSTM of a tence for ResNet- 101.	MS fea- tures Linear Bidi- ally ground- sen- tence re- sen- tence sen- representation.	Word embeddings COCO are projected to a ground space with a linear mapping. Linear mapping and Bi-LSTM are trained jointly. Three methods to ground sentences in images, captions or both:- Cap2Img: predict latent features of an image from its caption by mapping the (final) hidden state h(T) of the Bi-LSTM to the latent representation of the image. A ranking loss is to be minimized- Cap2Cap: given the caption pair (x,y) describing the same image, the goal is to maximize the joint probability of y given x. Negative log-likelihood as loss.- Cap2Both: Goal is to minimize the two loss functions above.In another setting, grounded and sentence-only (Skipthought) representations are concatenated with layer normalization to get the final sentence	Intrinsic- evalu- ation Skip the text- dings are of only word baseline embeddings:- MEN- Sim- Lex 999- Rare Words- WordSim- 353Extrinsic evaluations:- Movie re- view Senti- ment (MR)- Product re- views (CR)- Sub- jec- tivity classi- fica- tion (SUBJ)- Opin- ion polar- ity (MPQA)- Para- phrase iden- tifica- tion (MSRP)-

LM-Visual Lang Yer Paper	Pagele- mode Paper	IMG- Pre- trainings (LM)sour (LM)	IMG- Multimodal training sourc model	Testset/ ether tuning model	Baseline(s)/model set- tings/comparison to Testset/ ether tuning modelResults
2020 Bordes, Patrick et al. “In- cor- po- rat- ing vi- sual se- man- tics into sen- tence rep- re- sen- ta- tions within a grounded space.” arXiv preprint arXiv:2002.02734 (2020).	Tor Skipth et Corsual pusle- 11M book 74Ma or- pre- derained sen-In- tenc 13 words per net- sen-work tend on et average. 2016).	Processing Book with COCO 118K/5K 10K image Lt- Lg, which among its parameters has also those of the textual objective, which in turn profit from both objective functions.Lg is not applied directly on the sentence embeddings; it is trained on an intermediate space called the “grounded space”. The sentence embeddings are projected to the grounded space with the projection function being a multi-layer perceptron. The goal is to move away from the 1:1 correspondence between textual and visual space.Lg the can be decomposed in two components, whose individual contribution is controlled by two hyperparameters:- Cluster Information (Cg): sentences associated with the same image(s) should be similar. The visual space is the same as the textual space.	The objective function is composed of:- a (train/test) objective image Lt- a grounding book Lg, which among its parameters has also those of the textual objective, which in turn profit from both objective functions.Lg is not applied directly on the sentence embeddings; it is trained on an intermediate space called the “grounded space”. The sentence embeddings are projected to the grounded space with the projection function being a multi-layer perceptron. The goal is to move away from the 1:1 correspondence between textual and visual space.Lg the can be decomposed in two components, whose individual contribution is controlled by two hyperparameters:- Cluster Information (Cg): sentences associated with the same image(s) should be similar. The visual space is the same as the textual space.	Intrinsic- evalu- ation of word embeddi- ngs STS- SICK- Ex- trin- sic evalua- tions sic evalua- tions Movie re- view Senti- ment (MR)- Prod- uct re- views (CR)- Sub- jec- tivity classi- fica- tion (SUBJ)- al. (2017) Opin- ion polar- ity (MPQA)- Para- phrase iden- tifica- tion (MSRP)- Senti- ment classi- fication	Word Skipth embd- (text- dings only are bet- ter than embeddi- ngs) Foe textual bench- mark for evalua- tions with a high Kilos level of concrete- ness and are similar in perf- ormance with respect to more abstract al. (2015) Projects on the cross- modalspace are more effective than cross- projection sequential concatenation. concatenation always best per- formance on entail- ment tasks (bench- marks SNLI, SICK). Not best per- formance on entail- ment tasks (bench- marks SNLI, SICK).

Year	Language	Model	Pre-training	Multimodal training	Presentation and source	Model description	Testset	Baseline(s)/model set	Comparison to	Fine-tuning	Results
2020	Tan et al.	BERTEngResNExMS	Language model	-	-	-	GLUE	BERT	Improvement over	-	-

Hao, but Wikipedia. COCO with visual supervision. Each token in a sentence obtains a corresponding image (voken) assigned from a finite set of images. The voken is the image which maximizes a Relevance Score Function between a token and all images in the aforementioned finite set of images. With this token-voken pairs a voken classification pre-training task is performed that can be built in pure language models alongside other pre-training tasks such as MLM or Next-Sentence Prediction.

Year	Language	Page	Pre-training	Multimodal training	Presentation and source	Model description	Testset	Baseline(s)/model set	tings/comparison to	ether tuning	Results
2021	English	1	BERT Pre-CNNed - Rong- base ver-(ResNetMS hang, with sion50) and a of to Aman-learned the ex- preet task- embedding Singh. specific vi- "Unit: vec- Mul- tor ti- (to modal cap- ture (to task- specific learn- ing with a uni- fied trans- former ti- fied ad- di- former ti- fied ceed- ings of the IEEE/ CVF In- si- ter- na- tional Con- fer- ence on Com- puter Vi- sion. 2021. to- ken sequence.in- for- ma- tion) is con- cate- ated	IMG- Pre- trainingts (LM)sourc(IMG)	IMG- Multimodal training presentation and model description	To both modalities is then applied a COCO domain agnostic transformer Vi- architecture. As Geno transformer takes the hidden states of either language or visual encoders map + trans- former en- coder to en- code the hidden states and cross attention is applied to the encoded input map to a set of hid- den states to which a task-specific head states. A learned task- task spe- cific vec- tor (to cap- ture ded task- specific sequence.in- for- ma- tion) is con- cate- ated	GLUE:- QNLI- QQP- MNLI- SST2.	ExtrinsicBERT Model evalu- a- tion.: baselinetask	(text- setting only (1), single task)	training, outper- forms all other settings and is comparable to the text-only baseline Model setting (3), domain- agnostic, multi- task training with shared decoder across modalities	BERT Model (text- setting only (1), single task)

LM-Visual Lang- mode YePaper	Pagele- trainings Paper	IMG- Pre- training (LM)sour (IMG)	Multimodal training and sourc model	Testset/ ether tuning model	Baseline(s)/model set- tings/comparison to Results
2018 Shahmeham Has- san, Hen- drik, Lensch, R., Har- ald, Baayen. “Learn- ing zero- shot mul- ti- faceted vi- su- ally ground- word em- bed- dings via multi- task train- ing.” arXiv preprint arXiv:2104.07500 (2021).	Pre-trained GloVe, er-vec- siontors 2.2M of ob- words embeddings and fast- fer- ring 2M. penul- ti- mate layer of pre- trained Inception- V3 trained on Ima- geNet. A neu- ral net- work with one hid- den tanh acti- va- tion is used to project im- age vec- tors into the . . .	MS COCO originating from a pretrained text-only model, the goal is to generate a mapping matrix M to ground word embeddings visually (the mapping matrix is used in both directions, to map text to grounded space and to map grounded embeddings back to the textual space) This is obtained by performing three different tasks: (i) Next word prediction with a GRU, given previous words in the sentence provided as image caption, together with the related image embedding vector (ii) Same as (i) but the sentence is provided backwards to another GRU (iii) Binary classification task if the representation of a given sentence in the grounded space obtained from (i) and (ii) matched the associated image.	Given embeddings originating from a pretrained text-only model, the goal is to generate a mapping matrix M to ground word embeddings visually (the mapping matrix is used in both directions, to map text to grounded space and to map grounded embeddings back to the textual space) This is obtained by performing three different tasks: (i) Next word prediction with a GRU, given previous words in the sentence provided as image caption, together with the related image embedding vector (ii) Same as (i) but the sentence is provided backwards to another GRU (iii) Binary classification task if the representation of a given sentence in the grounded space obtained from (i) and (ii) matched the associated image.	Limited - to in- train- sic MEN- SimLex999- Rare Words- MTurk771- WordSim353- SimVerb3500. al. (2015) Park it with & real-world Myaengulations (2017) from the images Kiros (similar- ity al. (2018) appears to be Kiela favoured et by the al. (2018) del- over relatedness) Embeddings related to less concrete words exhibit good quality compared to baselines.	Textual GloVe baselines (text- and only related evaluations) models are outper- formed (text- and the only model MTurk771 ns to improve the textual vector space by aligning it with & real-world Myaengulations (2017) from the images Kiros (similar- ity al. (2018) appears to be Kiela favoured et by the al. (2018) del- over relatedness) Embeddings related to less concrete words exhibit good quality compared to baselines.

Year	Language	Model	Task	Testset	Baseline(s)/model set
2022	English	LM-Visual	IMG-PaG-e	Pre-training	Multimodal presentation and tuning

Su, Chan- Jan, Hung- yi, Lee, and Yu, Tsao. “XD- BERT: Dis- till- ing Vi- sual In- for- ma- tion to BERT from Cross- Modal Sys- tems to Im- prove Lan- guage Un- der- stand- ing.” arXiv preprint arXiv:2204.07316 (2022).

Pagele- mode training (LM) Paper (LM) source (IMG)

GLIP. WikiaText-2. ELECTRA. not specified. image- matching. ing sys- tem: two com- ponents, a text en- coder (CLIP- T) and an im- age en- coder (CLIP- ViT).

not specified. Language Modelling (MLM)- Same Sentence Prediction (MATCH)- CLIP concatenation with cross-modal encoder is performed.

3 adaptive tasks:- Masked Language Modelling (MLM)- CLIP Token Classification After that, concatenation with cross-modal encoder is performed.

- - Better GLUE- SWAG- READ.

BERT performance in pure language models, in particular in smaller datasets, which suggests that visual inputs improve generalization when the amount of training data is limited.

ELECTRA pure language models, in particular in smaller datasets, which suggests that visual inputs improve generalization when the amount of training data is limited.

Year	Language	Model	Training	Presentation	Description	Testset	Baseline(s)/model set
2022	Chinese	RoBERTa	Pre-trained	Multimodal	The framework is composed of two modules: 1.) Imagination generator G: for each text input, VQGAN generates an “imagined visual” and CLIP is used to test how the generated image corresponds to the text and encodes the text and the image in a cross-modal embedding space and the objective function Lgan is to minimize the distance between these two embeddings. 2.) Imagination augmented cross-modal encoder. Specifically, CLIP is used, with the embeddings from the textual and visual encoder (fed with the visualized imaginations) within CLIP are then “late fused”. The output is a set of “imagination-augmented” language representations. Learning procedure: 1.) pre-training of a visually-supervised transformer following the Vokenization	From GLUE, SST-2, QNLI, QQP, MultiNLP, MRPC, STS-B, FFocus	- Better BERT performance (textual baseline)

LM-Visual	IMG-		Baseline(s)/model
Lang	Pagele-	Pre-	set-
Paper	model	multimodal	tings/comparison
(LM)sour	trainings	training	to
YEPaper	(LM)	presentation and	Testset/ Ether
		sourcemodel	tuning modelResults

3.4 Text supporting computer vision models

Author: Max Schneider

Supervisor: Jann Goschenhofer

3.4.1 Introduction

“The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. [...] Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. [...] One thing that should be learned from the bitter lesson is the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great.”

— Sutton (XXXX)

This insight seems to directly inspire most model choices presented in this

chapter. Each network can be seen as an attempt of its company to employ their vast available resources on a large scale, with a particular focus on dataset sizes. These mostly become feasible through the adaptation of recent findings in natural language processing (NLP; see chapter 2.1) to computer vision (CV). On the one hand, architectural concepts firstly popularized in NLP are translated to CV (e.g., self-supervised learning or the Vision Transformer; ?) (see chapter 2.2). On the other hand, these powerful new NLP models, mostly Transformers (Vaswani et al., XXXXb), support bigger models from the inside as text encoding building blocks; hence the name of this chapter. There are subsections on the recent and relevant CV models CLIP (Radford et al., XXXXa), ALIGN (Jia et al., XXXXb) and Florence (Yuan et al., XXXXa) and a discussion of some relevant core concepts. The strong performances confirm the potential, hinted at by the impressive GPT-3 (Brown et al., XXXXa), of improving CV and increasing scale with the help of NLP.

3.4.2 Concepts

3.4.2.1 Web-scale data

A core problem that troubles researchers is the lack of robustness of previous state-of-the-art CV models to distribution shifts. I.e., when a model with good performance on its original dataset fails to generalize (transfer its knowledge) to new, more or less similar datasets. E.g., Radford et al. (XXXXa) report that a ResNet101 which they trained on ImageNet to an accuracy of 76.2% maintains only an accuracy of 32.6% on ObjectNet. This suggests that the model perhaps did not learn high quality latent representations, but instead overfit to the dataset-specific data-generating distribution. A common way to tackle this would be to try out various changes on the architecture and the training algorithm of the network. But this kind of adaptation, inscribing expert knowledge into the model, seems to repeat the mistake pointed out by Sutton (XXXX); “micromanaging” a model is likely to thwart future scaling.

The researchers of CLIP, ALIGN and Florence follow a different approach, based on scale. They try to increase sample size as much as possible and work with tremendous numbers of observations:

- 400 million (CLIP; Radford et al., XXXXa)
- 900 million (Florence; Yuan et al., XXXXa)
- 1.8 billion (ALIGN; Jia et al., XXXXb)

These are achieved with the vast amount of image-text pairs produced by and readily available on the internet. Thus, error prone, cost and labor intensive (difficult to scale), manual labeling is avoided. Unfortunately, the models trained on web data also become vulnerable to their downsides. Because of the

extremely noisy nature of them, still some form of pre-processing is needed, e.g., filtering for English language, excluding graphic content and, optionally, removing images with non-informative alt-texts. This makes some degree of dataset curation, and therefore arbitrary choices, necessary. Likewise, the social biases inherent to the internet are reproduced and furthermore, while the approach improves data efficiency to some degree (see next subsection 3.4.2.2), the poor performance of deep learning in this area is not substantially enhanced and mainly just compensated for with a super scalable source of supervision (Radford et al., XXXXa).

3.4.2.2 Contrastive objective

This source of supervision is the information contained in the co-occurrence of the image with its alt-text. It is accessed through natural language supervision. The architectures jointly train two sub-networks for image and text encoding, respectively. During this, the vector encodings are aligned in the latent representation space through minimizing a variant of the contrastive loss function (3.10) (?). Half of the first image-text pair loss

$$\ell_1^{V_{\text{img}}, V_{\text{txt}}} = - \mathbb{E}_{\{v_{\text{img}}^1, v_{\text{txt}}^1, \dots, v_{\text{txt}}^N\}} \left(\log \frac{h_\theta(\{v_{\text{img}}^1, v_{\text{txt}}^1\})}{h_\theta(\{v_{\text{img}}^1, v_{\text{txt}}^1\}) + \sum_{k=2}^N h_\theta(\{v_{\text{img}}^1, v_{\text{txt}}^k\})} \right), \quad (3.10)$$

where v_{img}^1 and v_{txt}^1 are vector encodings (latent representations) of image 1 and text 1 and $h_\theta(\cdot)$ is a similarity measure. In order to guarantee symmetry, the total loss is formed by the sum of $\ell_1^{V_{\text{img}}, V_{\text{txt}}}$ and $\ell_1^{V_{\text{txt}}, V_{\text{img}}}$, where the pairwise similarities of one text and every image is calculated instead of the other way around.

Figure 3.24 visualizes this. Initially all images and texts in the training data are encoded by the responsible sub-network. Using the resulting encodings, a similarity matrix with elements $h_\theta(\{v_{\text{img}}^i, v_{\text{txt}}^j\})$ can be calculated. Loosely speaking, the contrastive objective is to maximize elements on the diagonal and minimize the others.

Contrastive learning can be contrasted with classical predictive learning. Figure 3.25 gives an interesting insight into the choice of space, where goodness of fit is measured. The exemplary task is to color an image given its B/W version. Approach (a) first encodes the B/W image and then decodes the interim latent representation to fitting colors. The goodness of this fit is measured in the output space, meaning the estimated colors are compared to the true colors. Conversely, approach (b) measures the loss in the representation space.⁶ A reason for the good performance of contrastive learning could be

⁶Note that contrastive learning easily works with other combinations of modalities than text and image; here B/W and colors.

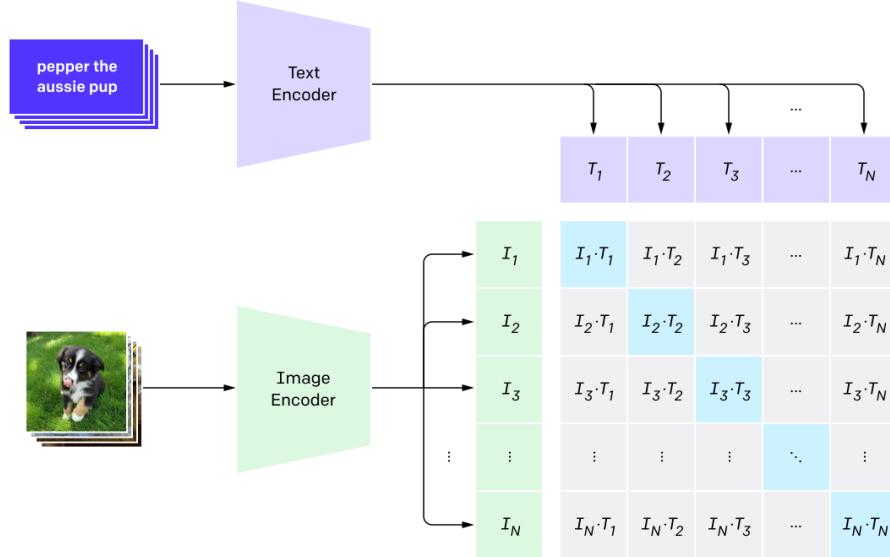


FIGURE 3.24: Visualization of a contrastive objective (Radford et al., XXXXa). After encoding the data, a similarity matrix for the images and texts is computed. The aim is that the N true image-text pairs score high in terms of similarity, while the $N^2 - N$ other possible combinations score low.

that, while common prediction losses (e.g., the \mathcal{L}_2 loss) penalize each prediction output dimension independently, approach (b) implies measurement in the intertwined representation space (?).

But in the end, rather than theoretical considerations, the driving factor for using this objective is data efficiency. As can be seen in figure 3.26, Radford et al. (XXXXa) start their search for an adequate pre-train model (more on this in subsection 3.4.2.3) by experimenting with a Transformer-based language model predicting the exact captions of an image. It turns out that this approach trains three times slower, in terms of data efficiency, compared to a simpler baseline of predicting a bag-of-words text encoding. Additionally, switching to the contrastive objective of CLIP improves data efficiency by a factor of four.

Nonetheless, the switch to contrastive learning leads to some limitations. Its rigidity demands certain extra steps and forfeits the high flexibility of generative models. In particular, this means contrastive models similar to CLIP are limited to choose from available options and cannot freely generate texts or images. To extend the capabilities of those models additional network building blocks are necessary.

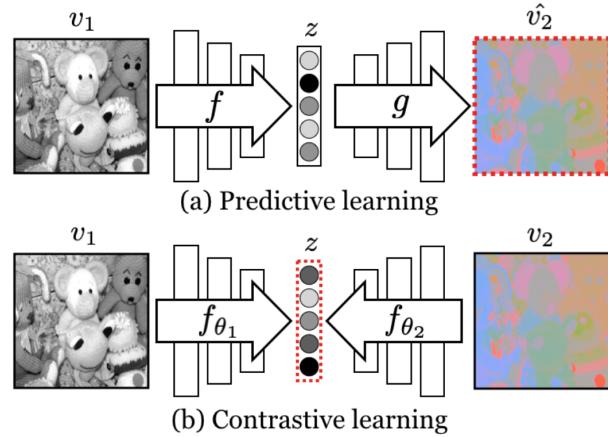


FIGURE 3.25: Predictive vs. contrastive learning: Predictive losses are measured in the output space while contrastive losses are measured in the representation space, indicated by red dotted boxes (?).

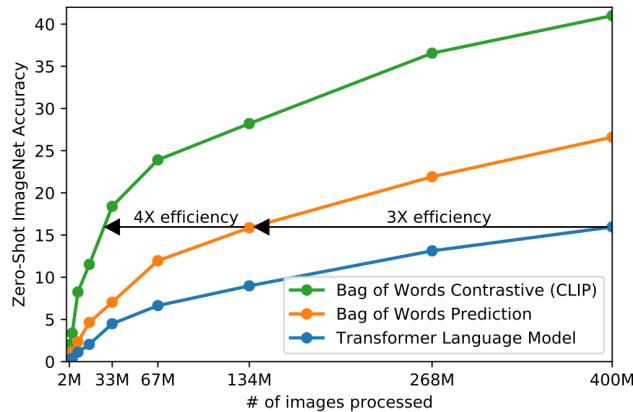


FIGURE 3.26: Data efficiency of contrastive objective. Development of zero-shot accuracy (see next subsection 3.4.2.3) on ImageNet with increasing number of instances of training data seen by models. The contrastive objective reaches same accuracies as the generative approach with only a seventh of the amount of data (Radford et al., XXXXa).

3.4.2.3 Foundation models and zero-shooting

The first models which are considered foundation models today began to appear in NLP. The term, later coined by ?, refers to models that are noteworthy due to their large scale and ability to adapt to a wide variety of downstream tasks. An early example is BERT (Devlin et al., XXXXb). A lot of the time, foundation models have an unfinished touch to them and the true scope of their capabilities cannot be sketched out clearly. This generally is the case, because the desired abilities of neural networks are not designed for explicitly, but rather emerge during their implementation. ? cite GPT-3's ability to perform certain types of new tasks solely by confronting it with the right natural language prompt. E.g., it is possible to get GPT-3 to summarize a paragraph by appending "TL;DR" (too long, didn't read) to the prompt, which is a common pattern on the internet to signal a following summary. This is called "in-context learning" (Brown et al., XXXXa). It is apparent that one can make up plenty of unexpected ways to employ these models and it cannot be known whether there is a further way no one thought of yet. This means possibly saving computational and data collection costs down the line, which ineptly is true for malicious use cases, e.g., surveillance, too.

Foundation models build on the concept of transfer-learning, i.e., pre-training a model on a task that is feasible to do and applying it to the desired task downstream. In the context of this chapter this means pre-training on web-scale data (see subsection 3.4.2.1) and evaluating performance on various common classification datasets. E.g., Radford et al. (XXXXa) name the SVHN dataset as a proxy for the task "street number transcription" with the caveat "on the distribution of Google Street View photos", but they remark that a lot of datasets have no obvious, specific task associated, e.g., CIFAR-10. They use these kind of datasets for measuring the "robustness to distribution shift and domain generation" of their model, which still is a topic of great interest as mentioned in subsection 3.4.2.1. When there is no further fine-tuning on the downstream task, i.e., no resuming of training on the new dataset, this is called zero-shooting. It has the clear advantage of evaluating performance more unbiased, as processes like overfitting to the data-generating distribution will not distort results.

Figure 3.27 shows how contrastive models perform zero-shot transfer. In the case of image classification all available classes are encoded by the language model. Afterwards, the CV sub-network computes the encoding of the image to be classified and all pair-wise similarity scores are returned. The pair with the best score can be retrieved as the decision. Image retrieval works the other way around: After an initial encoding of all images, the ones most similar to the encoded natural language text prompt in the representation space can be returned.

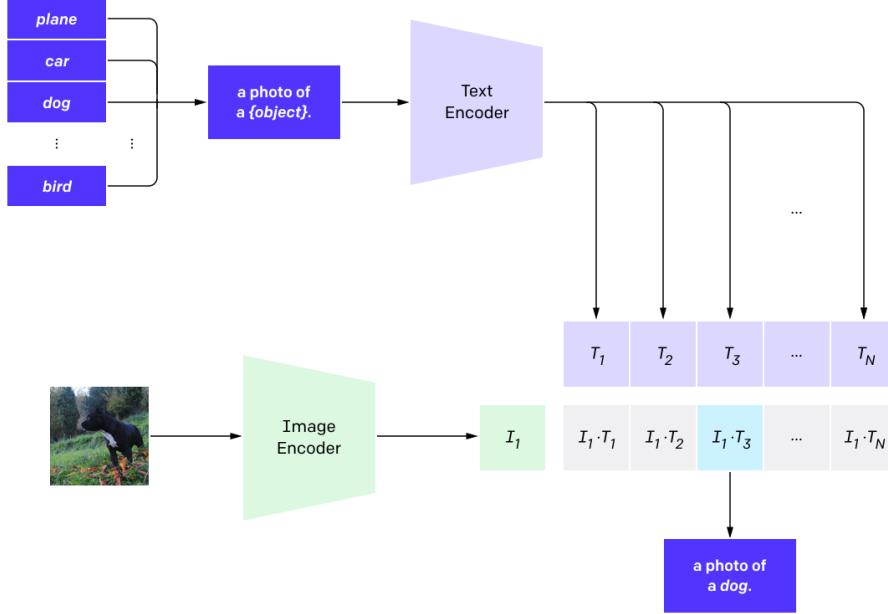


FIGURE 3.27: Visualization of zero-shooting (Radford et al., XXXXa).

3.4.3 Architectures

3.4.3.1 CLIP

The first of the large scale contrastive CV models that were published is CLIP, short for Contrastive Language-Image Pre-training (Radford et al., XXXXa). The components of its name are explained in previous subsections 3.4.2.2, 3.4.2.1 and 3.4.2.3 and are the crucial concepts of ALIGN and Florence as well. CLIP is a product of OpenAI, but its code is freely available and the different versions can be accessed as [python modules](#). Not released is its dataset.

A lot of preliminary work stems from Zhang et al. (XXXXb), whose paper firstly describes contrastive representation learning using image-text pairs. Their implementation of the contrastive loss function (3.10) is

$$\ell_1^{V_{\text{img}}, V_{\text{txt}}} = -\log \frac{\exp(\langle v_{\text{img}}^1, v_{\text{txt}}^1 \rangle / \tau)}{\sum_{k=1}^N \exp(\langle v_{\text{img}}^1, v_{\text{txt}}^k \rangle / \tau)}, \quad (3.11)$$

where $\langle v_{\text{img}}^1, v_{\text{txt}}^1 \rangle$ represents the cosine similarity, i.e., $v_{\text{img}}^{1\top} v_{\text{txt}}^1 / (\|v_{\text{img}}^1\| \|v_{\text{txt}}^1\|)$, and $\tau \in \mathbb{R}^+$ is a temperature parameter, which is directly learned during training (Zhang et al., XXXXb). CLIP adopts this. $\ell_1^{V_{\text{txt}}, V_{\text{img}}}$, the counterpart to $\ell_1^{V_{\text{img}}, V_{\text{txt}}}$ for the total loss, is function (3.11) with switched arguments. This

can be viewed as a symmetric cross entropy loss over the cosine similarity of the embeddings (Radford et al., XXXXa).

Architecture

The text encoder for CLIP (see figure 3.25) is a modified Transformer (Vaswani et al., XXXXb), which was also used for GPT-2 (Radford et al., XXXXd). For the image encoder multiple sub-networks are tested:

- ResNets: ResNet-50, ResNet-101
- ResNets which follow EfficientNet-style model scaling: RN50x4, RN50x16, RN50x64
- Vision Transformers: ViT-B/32, ViT-B/16, ViT-L/14

The best performing sub-network was the ViT-L/14. In turn, they trained it for an additional epoch with higher resolution images (336px), denoting this version [ViT-L/14@336px](#). If not indicated otherwise, the performances of this version of CLIP are displayed. The EfficientNet-style ResNets use x4, x16 and x64 of the compute of a ResNet-50 and the largest model (the RN50x64) trained for 18 days on 592 V100 GPUs, while the ViT-L/14 only took 12 days on 256 GPUs. The high parallelization capabilities of Transformers seem to pay off.

When explaining zero-shooting initially (see subsection 3.4.2.3), a text processing step was skipped. As can be seen in figure 3.27, there is an additional operation before the labels are fed into the text encoder. In order to help the model understand the context of the words, the class labels are embedded in a sentence, e.g., “A photo of a {label}.”. This increases the models zero-shot accuracy on ImageNet by 1.3 percentage points (pp). When ensembling 80 different context prompts⁷ Radford et al. (XXXXa) improve ImageNet accuracy by an additional 3.5pp, which adds up to a total of nearly 5pp. The average performance gain across 36 datasets is reported to be 5pp. It is likewise possible to directly communicate visual concepts like “picture”, “macro”, “drawing” or even “dog” to the model.

Robustness

Figure 3.28 reports the performance of CLIP and a ResNet101, whose training on ImageNet was stopped at the point it reached the same accuracy as zero-shot CLIP. It can be deduced that the methods studied in the paper of Radford et al. (XXXXa) constitute an important step towards closing the robustness gap mentioned earlier (see subsection 3.4.2.1). While the performance of the ResNet101 deteriorates with datasets generated from more and more different data distributions compared to ImageNet, CLIP remains fairly accurate. Note that these findings have to be taken with a grain of salt. Because OpenAI

⁷Prompts like: “A photo of a big {label}.”, “A photo of a small {label}.” (Radford et al., XXXXa)

does not grant public access to their training data, independent parties cannot investigate these claims on their own. E.g., it has to be relied on the conclusions of their overlap analysis to rule out that CLIP has not seen biasing amounts of future test data during training.

	Dataset Examples						ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet							76.2	76.2	0%
ImageNetV2							64.3	70.1	+5.8%
ImageNet-R							37.7	88.9	+51.2%
ObjectNet							32.6	72.3	+39.7%
ImageNet Sketch							25.2	60.2	+35.0%
ImageNet-A							2.7	77.1	+74.4%

FIGURE 3.28: Robustness of zero-shot CLIP to distribution shifts (Radford et al., XXXXa).

CLIP as a building block

Shen et al. (XXXX) study how the performance of Vision-and-Language (V&L) models improves, when the visual encoder is switched to CLIP’s strong image encoder. They discover that in this field of CV the ViT-B scores significantly worse than the ResNets. E.g., tests on image captioning reveal that the V&L model using ViT-B often performs only half as strong as the version using the RN50x4 (the largest network used in this study). This is possibly due to the pooling strategies of ViT-B, which result in a lack of visual localization abilities. Shen et al. (XXXX) test their hypothesis and generate, e.g., figure 3.29 which depicts Grad-CAM Visualizations for a V&L model with a ViT-B backbone and a ResNet-50 backbone and the question “What color is the woman’s shirt on the left?”. The red area indicates relevant pixels and appears much more focused for CLIP-Res50 than for CLIP-ViT-B.

3.4.3.2 ALIGN

The approach of Jia et al. (XXXXb) is largely similar to CLIP. They reiterate the necessity of large-scale vision datasets, but assert that even CLIP’s data collection process still involves a non-trivial amount of data curation. They propose that the amount of additional observations obtained through mini-

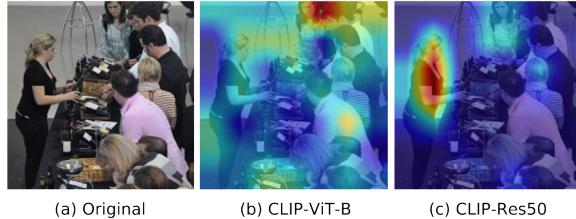


FIGURE 3.29: Grad-CAM Visualizations for the prompt “What color is the woman’s shirt on the left?”.

mizing the amount of filtering makes up for the increased noise. The result is a dataset with 1.8 billion image-text pairs. The corresponding model is named ALIGN, short for “A Large-scale ImaGe and Noisy-text embedding”, whose acronym hints at the contrastive loss, which aligns vector encodings in the representation space (see subsection 3.4.2.2).

Architecture

ALIGN follows the dual encoder architecture employed by Zhang et al. (XXXb) and Radford et al. (XXXa), but uses part of BERT-Large as the text and EfficientNet-L2 as the image encoder, which they jointly train from scratch. The model has around 800 million parameters (Alford, XXXX). Subsection 3.4.4 goes into more detail about the performance of ALIGN and compares all three models discussed in this subsection.

Connecting image and text representations

The contrastive loss function aligns the latent representations of the different modalities. In other words, the explicit objective is that similar vector encodings implicate similar inputs. This means arithmetic operations like the ones mentioned in chapter 2.1 are not only meaningful on encodings belonging to the same modality, but to different modalities. E.g., one can add up the image encoding of a picture of the Eiffel tower and the text encoding of the word “snow” and retrieve pictures with high cosine similarity to the result, see figure 3.30.

3.4.3.3 Florence

While in principle the approach of Yuan et al. (XXXa) does not largely differ from the others, the focus of this paper is more about making a true foundation model. In order to achieve this, they propose a map of possible vision applications which they try to cover via extending the core model with modules. As figure 3.31 depicts, they want to advance into the dimensions of fine-grained object detection, dynamic action recognition and true multimodal

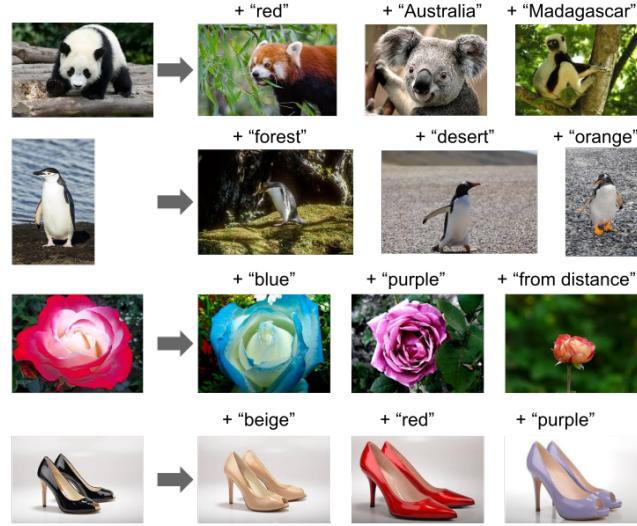


FIGURE 3.30: Multimodal image retrieval via arithmetic operations on word and image embeddings.

tasks. Because of their big ambitions, they name their model Florence after “the birthplace of Renaissance” (Yuan et al., XXXXa).

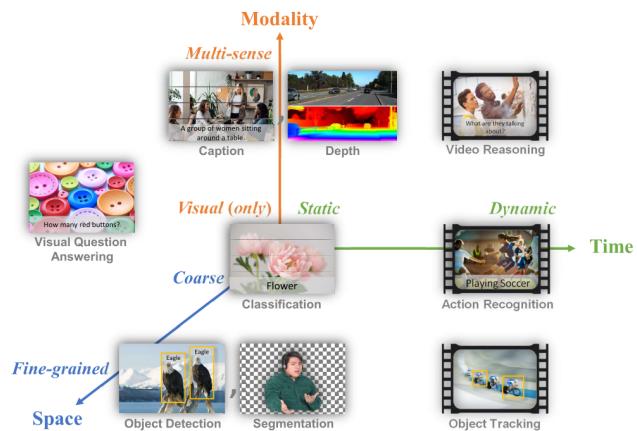


FIGURE 3.31: Florence’ approach to foundation models: A general purpose vision system for all tasks.

Architecture

For the two encoders of the pre-trained core they use a hierarchical Vision Transformer (CoSwin Transformer) for images and a Transformer similar to

CLIP's for text. Their 893 million parameters are also jointly trained from scratch on 900 million image-text pairs. The alignment happens in the so called image-label-description space which is encoded through a special version of the contrastive loss function which regards all image-text pairs with the same label as positive instances. Figure 3.32 depicts their version of figure 3.24 where one can schematically see how they flexibly add modules to the pre-trained core in order to adapt to various downstream tasks.

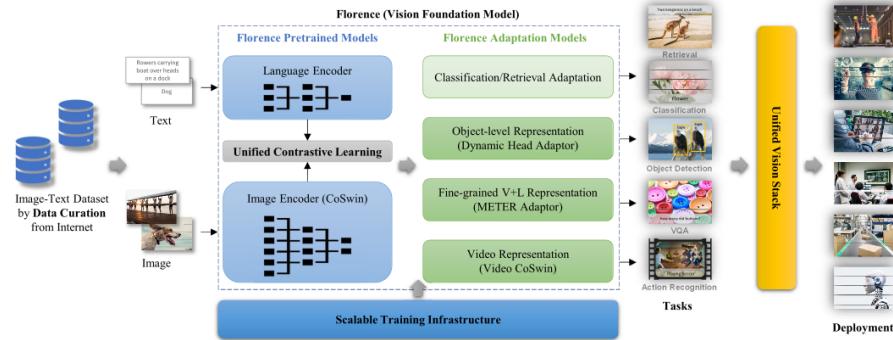


FIGURE 3.32: Modular architecture of Florence.

3.4.4 Performance comparison

Throughout the papers of Radford et al. (XXXXa), Jia et al. (XXXXb) and Yuan et al. (XXXXa) it was possible to collect three tables with reported performance measures.

Table 3.33 summarizes the zero-shot accuracies on four different ImageNet variants. Unfortunately Yuan et al. (XXXXa) only stated their performance on the original ImageNet, where they beat CLIP and ALIGN by a margin of 7.3pp. The results on the other three ImageNet pendants are mixed and there is no clear winner between CLIP and ALIGN.

	ImageNet	ImageNet-R	ImageNet-A	ImageNet-V2
CLIP	76.2	88.9	77.2	70.1
ALIGN	76.4	92.2	75.8	70.1
Florence	83.7			

FIGURE 3.33: Top-1 Accuracy of zero-shot transfer of models to image classification on ImageNet and its variants.

Table 3.34 concerns zero-shot image retrieval on the Flickr30K and the

MSCOCO dataset (see chapter 2.3). Even though there are not many major score differences, there is a clear ranking with CLIP on third, ALIGN on second and Florence on first place.

	Flickr30K (1K test set)				MSCOCO (5K test set)			
	Image→Text		Text→Image		Image→Text		Text→Image	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP	88.0	98.7	68.7	90.6	58.4	81.5	37.8	62.4
ALIGN	88.6	98.7	75.7	93.8	58.6	83.0	45.6	69.8
Florence	90.9	99.1	76.7	93.6	64.7	85.9	47.2	71.4

FIGURE 3.34: Zero-shot image and text retrieval (Yuan et al., XXXXa).

The most comprehensive comparison is comprised in table 3.35. It depicts the accuracy of zero-shot CLIP and Florence on various datasets as well as the scores of all three models fine tuned to the respective datasets. Florence beats CLIP in nearly all evaluations, for the zero-shot setting as well as for fine tuned performance. Jia et al. (XXXXb) only report on four of the twelve dataset, where they lead half of the time.

	Food101	CIFAR10	CIFAR100	SUN397	Stanford Cars	FGCV Aircraft	VOC2007	DTD	Oxford Pets	Caltech101	Flowers102	ImageNet
	CLIP	93.8	95.7	77.5	68.4	78.8	37.2	84.3	55.7	93.5	92.8	78.3
Florence	95.1	94.6	77.6	77.0	93.2	55.5	85.5	66.4	95.9	94.7	86.2	83.7
CLIP (fine tuned)	95.9	97.9	87.4	82.2	91.5	71.6	89.9	83.0	95.1	96.0	99.2	85.4
ALIGN (fine tuned)	95.9								96.2			88.6
Florence (fine tuned)	96.2	97.6	87.1	84.2	95.7	83.9	90.5	86.0	96.4	96.6	99.7	90.1

FIGURE 3.35: Top-1 Accuracy of CLIP, Florence and ALIGN on various datasets.

Summing up, ALIGN achieves its goal of replicating CLIP’s impressive performance while dramatically reducing data curation efforts and Florence has the overall top performance. This could be attributed to its custom loss, maybe to Yuan et al. (XXXXa) striking the best balance between sample size and data curation or to Florence having the best sub-networks; or a combination of all three.

Once again note that none of the training datasets were made publicly available. It cannot be guaranteed that all benchmarks were evaluated on unseen datasets.

3.4.5 Resources

One can access the pre-trained CLIP models on [Github](#) and they even found their way into simple command line tools already. For example there is a CLI named [rclip](#), which can be used for personal image retrieval, wrapping the *ViT-B/32* CLIP architecture. On my (mid-range) laptop I was able to find seemingly good matches for search terms which I tried out inside a folder containing about 100 pictures. After an initial caching one request took about ten seconds. Furthermore CLIP continues to be used inside new models, e.g., DALL·E 2, where it is responsible for image embedding (?). Also, there is an effort to replicate CLIP's training dataset called LAION-400M ([Schuhmann, XXXX](#)). To validate the image-text pairs collected for this, their cosine similarity is computed using CLIP and instances with a value too low are discarded. As to my knowledge nothing was made publicly available as part of the other two papers.

3.5 Text + Image

Author: Steffen Jauch-Walser

Supervisor: Daniel Schalk

Data is naturally at the heart of every data scientific issue. While there have been many advances made in machine learning in recent years, many promising research areas remain, as do a multitude of problems associated with them. One such promising area are multi-modal machine learning models. Combining different input data is a key aspect towards making models more sophisticated. When thinking about teaching robots specific tasks, detecting hateful memes or deep fakes, it is apparent that only through the combination of multiple modalities, success might be achieved. Context is key.

However, learning context requires increasingly complex models. While early machine learning models built their success upon the possibility to analyze the big pool of available, often unstructured data, modern machine learning models are so demanding that there is often not enough data or training time available. Obtaining data is a major issue for multi-modal machine learning. Since labelling data in vast amounts is prohibitively expensive, larger models have to come up with specific strategies to move forward such as self-supervised training or automatically scraped web datasets. Nevertheless, when models become so large that billions of parameters have to be learned, even scraping the whole web starts to show its limits. Another natural issue is the transformation of different types of data into usable model inputs.

There is no shortage of different single modality machine learning models. On

the contrary, when every new hyperparameter configuration might be seen a new model, it becomes hard to keep track. More importantly, it is often not clear how a model from one area transfers to another. Did we learn some modality specific bias or a general principle? Consolidating different models into a unifying framework is a key prospect of multimodal machine learning. While the grand dream of a single unifying model might be out of reach, consolidating different areas is well in sight. In the following, we will have a look at the challenges and prospects of multimodal machine learning against the background of visual language models. Visual Language Models are models which can deal with both language and images as input data. Specifically, we will have a closer look at three different models: Data2vec, VilBert and Flamingo. Data2vec is an unsupervised model that can handle different modalities, but not their interaction, using a single unifying training framework. VilBert is an early visual-language model that can handle interactions between images and text through its innovative concept of cross-attention. Flamingo is a recent few shot visual language model that features large expressive text capabilities through the use of a large language model. With 80B parameters, it particularly highlights how to leverage the communication between frozen models when further scaling up the model size.

An overview across the popularity of current research fields in visual language modelling is provided in figure 3.36. A detailed list of trends for each of those fields can be found in the accompanying paper ([Uppal et al. \(XXXX\)](#)). Most research is done in the areas of visual question answering (VQA) and visual captioning (VC), but also for example visual commonsense reasoning (VCR), vision-language navigation (VLN) or multimodal affective computing (MAC). MAC uses images and text to infer sentiment, for example through facial expressions. VCR as an extension of VQA is particularly interesting in the realm of making models more interpretable. After all, we would like to know why machine learning models do what they do. Finally, VLN has many promising practical applications in the field of robotics, particularly the interaction of humans and robots.

3.5.1 Data2vec

With data2vec ([Baevski et al. \(XXXa\)](#)), data scientists at Meta, formerly Facebook, developed an architecture that addresses some of the mentioned issues while highlighting the importance of sophisticated training schemes. Their algorithmic structure is able to work with either text, image or speech data. On top of that, the model is self-supervised based on a teacher-student relationship which reduces the need for human labelling. It is not a universal model in the sense that it works with any input, nor is it even a general model in the sense that the algorithm is exactly the same for each modality. However, the overall model structure remains the same for either text, speech or image input data, while only the specific encoding, normalization and mask-

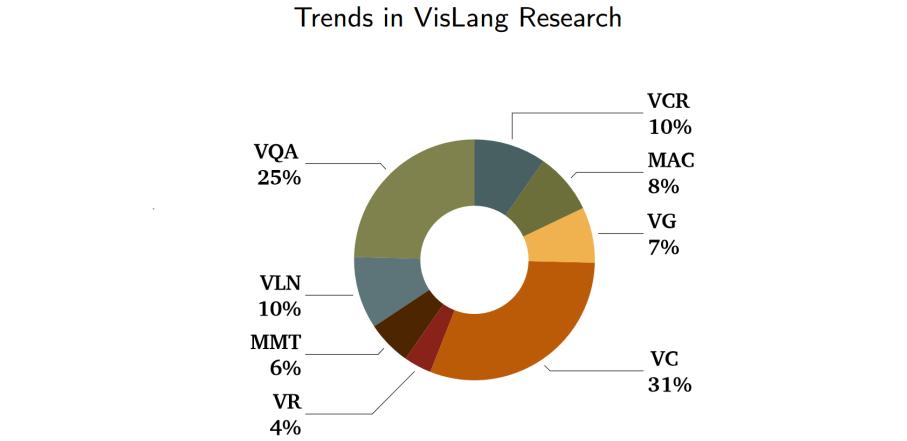


FIGURE 3.36: Uppal et al. (2022): VisLang Paper Trends (previous 2 years)

ing strategies are modality-specific. In that regard, it is a step towards a more general way of dealing with different modalities and it is very effective at doing so given the benchmark results on typical data sets. Particularly noteworthy is also the way they implement the self-supervised learning. Data2vec predicts contextualized and continuous representations rather than typically used discrete tokens such as sub-words. Working with latent representations of the input space has two advantages: not only is the number of prediction targets not a-priori limited, but they are also richer in information.

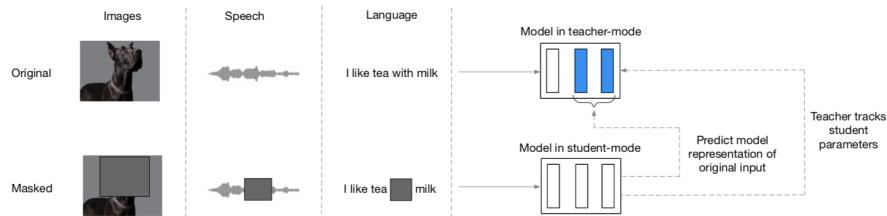


FIGURE 3.37: Baevski et al. (2022): Data2vec Architecture - a teacher model creates contextualized latent targets on the basis of its top K layers (blue) as prediction task to train the student model

Figure 3.37 depicts the general model architecture. The two main components are a teacher and a student model which only differ in one aspect, the weights of the teacher model are an exponentially decaying average of the student's weights. The purpose of the teacher model is to create training targets for the student model. In a first step, a modality is chosen and inputs are encoded

according to the specific encoding scheme for that modality. A masked version is given to student model, but notably, the teacher model has access to an unmasked, complete view of the input data. Hence, the resulting training targets will be fully contextualized using a self-attention mechanism over the whole input data. The training targets are based on the top K layers of the teacher model depicted in blue in 3.37 . More specifically, denote by y_t the training target at time t and by a_t^l the outputs of the l-th block, then $y_t = \frac{1}{K} \sum_{l=L-K+1}^L \hat{a}_t^l$, i.e. the training targets are the average of the outputs of the top K layers of the teacher network after a normalization has been applied. Normalization helps to stabilize the training process and prevent model collapse which can be an issue with models that learn their own representation.

From the authors point of view, working with a latent representation of the actual learner as training target is a simplification of many commonly used modality-specific designs despite the caveat that this paper still uses modality-specific encoding strategies. Compared to other models, there is no cross-modality training. The specific loss function used to regress the targets is a smooth L1 loss.

$$L(y_t, f_t(x)) = \begin{cases} \frac{(y_t - f_t(x))^2}{\beta} & \text{if } |(y_t - f_t(x))| \leq \beta \\ |(y_t - f_t(x))| - \frac{\beta}{2} & \text{otherwise} \end{cases}$$

Using a smooth L1 loss has the advantage of being continuous, yet sensitive to outliers, however the β parameter needs tuning. As far as the general model architecture is concerned, the underlying architecture is a standard transformer architecture ([Vaswani et al. \(XXXXb\)](#)).

How does the modality specific input handling work? In many ways, in this work the authors combine the strategies developed in multiple previous works and add a unifying framework on top of it. For images, the typical Vision Transformer (ViT) strategy (3.38) to transform images with a size of 224x224 pixels into 16x16 pixel patches is employed. Every patch is then linearly transformed into a sequence of 196 flattened representations including a learnable positional encoding that serve as input to the vision transformer. A classification token is used to produce the final categorization. The contextualization is produced in the multi-head attention blocks as explained in earlier chapters. In short, multi-head attention first projects the keys, queries and values with learned linear projections which are then evaluated in parallel to create more expressive attention maps. Attention itself is calculated as scaled dot-product-attention using a softmax over the scaled product of keys, queries and values. ([Vaswani et al. \(XXXXb\)](#)) As far as the vision transformer itself is concerned, datav2vec tests two different model sizes, a base model size of 12 and a large model of 24 transformer blocks. The masking strategy for images follows the Bert pretraining approach of image transformers, BEiT, proposed by ([Bao et al. \(XXXX\)](#)). In particular, multiple adjacent blocks are being masked with random aspect ratio. The minimum size of a masked block is 16 patches.

In total, 60% of patches were masked in the data2vec algorithm, which is an increase over the original 40% used by BEiT. However, the authors note that they found increased masking to be more accurate. The augmentation strategies are similar, as well. Resizing crops, horizontal flipping and colour jittering were used. Naturally, the student and teacher model are given the same modified image. Finally, for image data, the model is measured on a classification task. Hence, the authors use a mean-pooling over all patches in the last transformer block and input that into a softmax-normalized projection that conducts the classification, which is again based on the BEiT model.

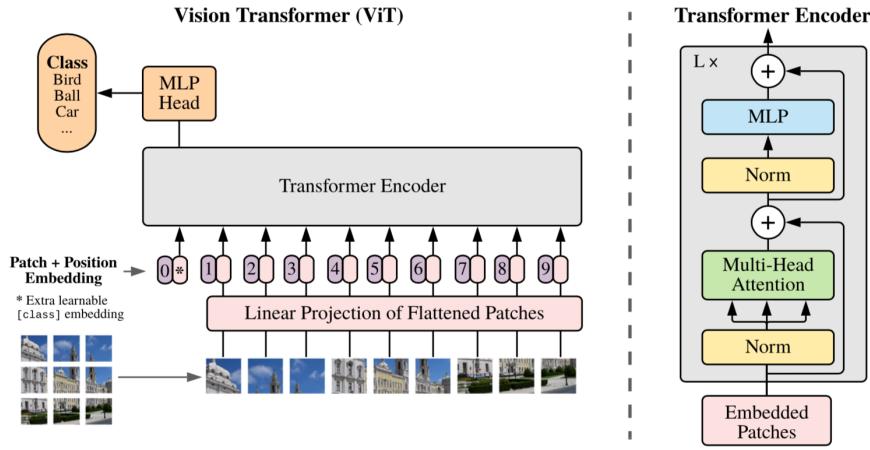


FIGURE 3.38: Dosovitskiy et al. (2021)

The natural language processing model is implemented with a PyTorch toolkit named fairseq and based on the RoBERTa (Liu et al. (XXXX)) architecture which redesigned the standard Bert model training procedure to make it more robust and effective. In particular, it increases hyperparameters such as the learning rate and the batch size. It also removes the next sentence prediction task to improve on the masked language modelling performance. In this case they follow (Sennrich et al. (XXXXa)) and encode sub-words as 50k byte-pairs. A separate embedding vector is learned for each type. For the masking, the Bert masking is being used. 15% of the embedded tokens are replaced, thereof 80 percent are learned masks, 10% are unchanged and the remaining 10% are replaced with random tokens in the vocabulary. Another strategy that the authors also consider is the wave2vec masking strategy to mask four consecutive tokens with a probability of 0.35 while only using learned tokens (Baevski et al. (XXXXb)). As it turns out, the later strategy further improves results. The natural language processing model is evaluated on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al. (XXXXa)) which

includes for example includes NLP inference, sentence similarity and sentiment analysis tasks.

The speech category is also implemented in fairseq. The feature encoder for speech is based on the wave2vec framework and uses 16kHz inputs. It is built upon seven temporal convolutions intertwined with normalization layers and a GELU activation function such that the output of the encoder is 50 kHz.

As far as the results are concerned, data2vec achieved state-of-the-art performance in vision and language tasks among similar self-supervised models.

	ViT-B	ViT-L
<i>Multiple models</i>		
BEiT (Bao et al., 2021)	83.2	85.2
PeCo (Dong et al., 2022)	84.5	86.5
<i>Single models</i>		
MoCo v3 (Chen et al., 2021b)	83.2	84.1
DINO (Caron et al., 2021)	82.8	-
MAE (He et al., 2021)	83.6	85.9
SimMIM (Xie et al., 2021)	83.8	-
iBOT (Zhou et al., 2021)	83.8	-
MaskFeat (Wei et al., 2021)	84.0	85.7
data2vec	84.2	86.6

FIGURE 3.39: Baevski et al. (2022): data2vec performance (vision)

Figure 3.39 shows the model's performance in computer vision. Pretrained and fine-tuned simply on the data of the well known ImageNet-1K dataset, data2vec was evaluated using top1-accuracy, the standard notion of accuracy, on the task to predict single labels for images. The base model ViT-B comprises 86M parameters and ViT-L 307M parameters. The results show that predicting contextualized latent representations in a masked prediction setup can work well as model training compared to classical local methods such as predicting visual tokens. MoCov3 (Chen et al. (XXXXb)) is a self-supervised model trained on a contrastive loss. The most similar model is DINO (Caron et al. (XXXX)), which also uses a self-distillation setup to predict teacher outputs using a cross-entropy loss. However, their prediction target was the final layer rather than averaged layers while using differing images for teacher and student network. The well performing MAE model (He et al. (XXXXa)) is a masked autoencoder which is trained on reconstructing masked pixels using an asymmetric encoder-decoder architecture. In contrast, MaskFeat (Wei et al. (XXXX)) uses masked feature prediction. Notably, data2vec outperforms all of them although trained for the same amount or less. Particularly, MAE and MaskFeat use 1600 epochs rather than 800 like data2vec.

Figure 3.40 shows the performance in the language domain. For the language domain, the model is evaluated on the GLUE benchmark (Wang et al. (XXXXa)). The model is pretrained and fine-tuned separately on the labelled

Table 3. Natural language processing: GLUE results on the development set for single-task fine-tuning of individual models. For MNLI we report accuracy on both the matched and unmatched dev sets, for MRPC and QQP, we report the unweighted average of accuracy and F1, for STS-B the unweighted average of Pearson and Spearman correlation, for CoLA we report Matthews correlation and for all other tasks we report accuracy. BERT Base results are from Wu et al. (2020) and our baseline is RoBERTa re-trained in a similar setup as BERT. We also report results with wav2vec 2.0 style masking of spans of four BPE tokens with no unmasked tokens or random targets.

	MNLI	QNLI	RTE	MRPC	QQP	STS-B	CoLA	SST	Avg.
BERT (Devlin et al., 2019)	84.0/84.4	89.0	61.0	86.3	89.1	89.5	57.3	93.0	80.7
Baseline (Liu et al., 2019)	84.1/83.9	90.4	69.3	89.0	89.3	88.9	56.8	92.3	82.5
data2vec	83.2/83.0	90.9	67.0	90.2	89.1	87.2	62.2	91.8	82.7
+ wav2vec 2.0 masking	82.8/83.4	91.1	69.9	90.0	89.0	87.7	60.3	92.4	82.9

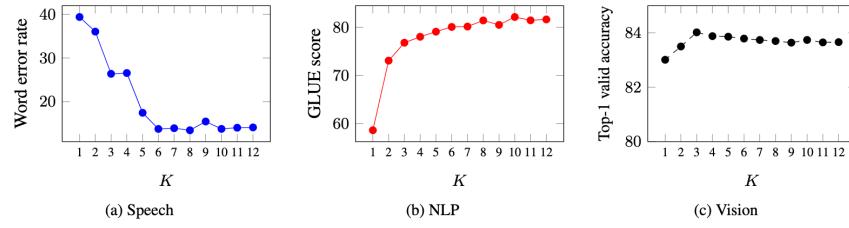


Figure 2. Predicting targets which are the average of multiple layers is more robust than predicting only the top most layer ($K = 1$) for most modalities. We show the performance of predicting the average of K teacher layer representations (§3.3). The effect is very pronounced for speech and NLP while for vision there is still a slight advantage of predicting more than a single layer.

FIGURE 3.40: Baevski et al. (2022): data2vec results (language)

data from each task. Accuracy is reported as the average across 5 tuning cycles. While data2vec achieves a higher average performance than the baseline model, there are tasks where the baseline model prevails. A large portion of the performance difference seems to be driven by the CoLA task. The Corpus of Linguistic Acceptability (CoLA) consists of 10657 sentences from 23 linguistics publications and the task is to judge whether they are grammatically correct. Hence, it is distinctly different from the other tasks. The Stanford Sentiment Treebank (SST) analyzes sentiment in language through movie reviews. The Multi-Genre Natural Language Inference (MultiNLI) corpus contains sentence pairs and focusses on textual entailment across genres. Similar tasks are used in the Recognizing Textual Entailment (RTE) dataset which focuses on text from news and wikipedia. The QNLI (Question-answering NLI) dataset is a Natural Language Inference dataset that contains answers from wikipedia to corresponding questions posed by an annotator. The task for the model is to find out whether the sentence contains the answer to the question. QQP stands for Quora Question Pairs, which analyzes paraphrases. Finally, the Microsoft Research Paraphrase Corpus (MRPC) also consists of sentence pairs from newswires which may or may not be paraphrases of each other.

As a suitable baseline model, the authors retrain RoBERTa in the respective setup. On top of the heterogenous performance across language tasks, the evaluation also clearly shows that averaging over multiple layers to create prediction targets improves performance across all three domains. The effects seem to be most pronounced on NLP tasks whereas CV does not benefit

from averaging more than three layers. In the speech domain, six layers seems to be enough to reach peak performance. In any case, performance loss while following the strategy to simply average the maximum amount of layers, rather than fine-tuning K, seems small enough to be potentially acceptable.

To sum it up, data2vec is a self-supervised model that can work with either text, speech or image data, but not across modalities. It aims at unifying the learning framework through a teacher-student-setup that allows for contextualized latent target prediction. The teacher model is based on a complete view of the input data, which introduces contextualization, while the student model only sees a masked version of the input. Compared to previous work, the authors average the top K layers rather than only the final layer of the model, which has a notable effect as shown in 3.40. As there are different layers in the transformer network, the authors also investigate which layers work best for prediction. They conclude that the output of the feedforward layer works best. Built on a transformer architecture, self-attention is the main driver that creates contextualized targets in the teacher model and hence performance. The authors also show that contextualization through the teacher model works best with the complete view of the input rather than a partial view. On top of not being able to work across modalities, one drawback is that the model's structure still uses modality specific encoding and masking schemes. In that regard, the perceiver architecture ([Jaegle et al. \(XXXX\)](#)) for example used in the Flamingo model is a complementary approach worth exploring. An earlier model that works across modalities is VilBert.

3.5.2 Vision-and-Language Bert (VilBert)

As seen in the previous section, data2vec can handle text, image or speech as input data. However, it cannot do so at the same time. The model's focus is on unifying the training approach rather than working across modalities. However, when we think about multimodal models, we usually think of working with different modalities at the same time. VilBert ([Lu et al. \(XXXXb\)](#)) is a natural extension of the iconic Bert architecture ([Devlin et al. \(XXXXd\)](#)) to vision-and-language modelling. An immediate question is whether vision and language inputs should be handled together in a single stream or in parallel. As we will see, it turns out that encoding inputs in parallel and working with parallel streams increases performance. At heart of that architecture is a co-attention mechanism which enables information exchange between both modalities.

Figure 3.41 shows the employed parallel stream architecture. Each modality is handled separately and fed into two Bert-style transformer models. This allows for both modalities to be handled according to their respective needs while co-attention layers allow for communication between the streams. For the language stream, the encoding uses the vocabulary plus a special classification

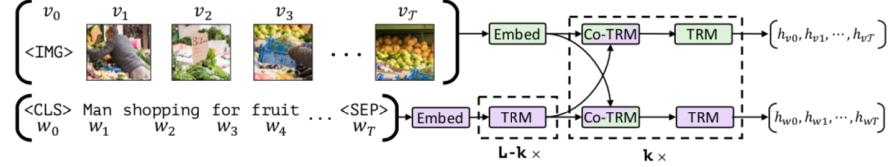


FIGURE 3.41: Lu et al. (2019): VilBert’s Dual Stream Architecture: dashed transformer modules can be repeated, co-attention modules allow sparse interaction between modalities.

token (cls), a sentence separation token (sep) and a masking token (mask). For the vision stream, image region features are extracted via a Faster R-CNN (Ren et al. (XXXX)) model which was pretrained on the Visual Genome Dataset (Krishna et al. (XXXX)). Since image regions lack a natural ordering, its spatial location has to be encoded, as well. VilBert achieves that through a five dimensional vector that encapsulates the image coordinates and the fraction of the covered image area. Through projection, the dimensions of the positional encoding and visual features are matched and then summed. The image token marks the beginning of such an image region sequence while representing the whole image.

Through the dual stream architecture, the complexity of the model can be adjusted separately for each modality. An alternative approach would have to discretize the visual space via clustering and then use the resulting tokens in the same way as text tokens. The drawbacks of that approach are the potential loss of detail at the discretization stage and the loss of flexibility across modalities as a result of the same processing. Finally, a single stream architecture can interfere with pretraining of the language models. The model will have to be fine-tuned based on the created visual tokens. As those might be very different from the text tokens, there is potential for the pretrained language model to become ‘damaged’ in the process and lose capabilities - and idea that is also central to the Flamingo model presented later on.

The key innovation in the Vilbert paper (Lu et al. (XXXXb)) is the use of co-attention layers. In figure 3.42, the basic architecture is depicted. The co-attention module computes query, key and value matrices in a standard transformer attention fashion. However, it then feeds the keys and values from each modality into the other modalities multi-head-attention block. As a result, the visual attention will be conditioned on text whereas the language attention will be image-conditioned. This communication between streams only occurs at specific sections in the model, denoted by co-trm in figure 3.41. Notably, the language stream features a lot more preprocessing before the first co-attention layer than the image stream.

An interesting question to ask is what is actually learned in those atten-

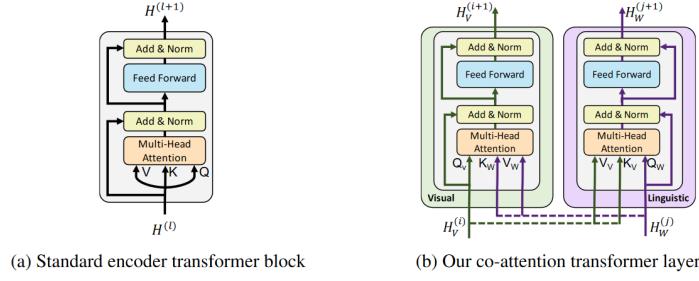


Figure 2: We introduce a novel co-attention mechanism based on the transformer architecture. By exchanging key-value pairs in multi-headed attention, this structure enables vision-attended language features to be incorporated into visual representations (and vice versa).

FIGURE 3.42: Lu et al. (2019): Cross-Attention in VilBert

tion layers and how they correspond to human attention maps. (Sikarwar and Kreiman (XXXX)) analyze the efficacy of co-attention layers for VQA tasks in a VilBert network. Specifically, they compute the question conditioned image attention scores and compare them to human attention maps created in experiments. In those experiments, humans are tasked with unblurring specific image regions to answer the same questions one would expect the machine learning model to answer. Such human attention maps are collected in the VQA-HAT dataset (Das et al. (XXXX)). Rank correlation is used to compare attention maps. (Sikarwar and Kreiman (XXXX)) find that in a 6 layer network rank correlation plateaus at layer 4 and increases in the number of image regions proposed while encoding the images. Perhaps more surprisingly, they find a minimal influence of semantics on the generation of the attention maps. Randomly shuffling words in a sentence when testing the model performance barely changes the attention output, which suggests that keywords rather than sentence structures drive the attention output. Note however that while attention maps remained similar, the model’s actual performance on answering the questions dropped noticeably by approximately 15% such that it seems clear that coherent sentences are important for the overall VQA task, but not for the attention creation process. What are the keyword that drive cross-attention in VilBert? The evidence provided by the authors clearly shows that nouns are the most influential parts-of-speech when considering attention maps. On top of that, prepositions can sometimes help identify spatial relations. There is also some support for the hypothesis that removing Wh-words such as “who” and “where” can improve fine-grained attention maps in the final layer which might be worth exploring further as preprocessing for deeper networks. Another approach would be to search for ways to improve the way attention maps are generated by finding ways to include more of the available sentence information. Most notably, however, using object-based region proposals to process images can lead to bottlenecks that can prevent the model

from learning sufficiently fine-grained attention maps as shown in figure 3.43. Overall, humans are naturally good at VQA tasks. Hence, it is not surprising that attention maps which correlate well with human attention maps also improve model performance.

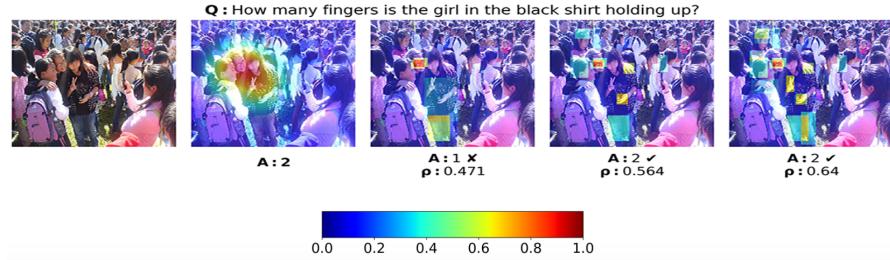


FIGURE 3.43: Sikarwar et al. (2022): (Left to Right) Picture, Human Attention, 36 Regions, 72 Regions, 108 Regions. Similarity between human and model attention is measured using rank correlation.

Figure 3.43 shows that the number of region proposals fed into the model after processing an image affects the ability of the model to produce adequate attention maps. In this particular case the question “How many fingers is the girl in the black shirt holding up?” was correctly answered by humans, as well as a VilBert model using 72 or 108 region proposals. It was answered incorrectly when using only 36 region proposals. Note however that in either case, the machine learning model captured the face of the wrong girl. The model using 72 regions also identified the wrong hand despite answering the question correctly. While the 108 region model identifies the correct hand holding up the fingers, it does not seem to prioritize it over the other identified hands in the picture. Hence, the attention maps are sufficiently different from the human attention map which highlights the need to look closer not only at how models are performing, but also into how their performance has been achieved.

As far as the model training is concerned, VilBert is pretrained and fine-tuned. The pretaining tasks comprise masked-multi-modal modelling and multi-modal alignment prediction performed on the Conceptual Captions dataset. That dataset contains about 3,1 million usable aligned image-caption pairs, which have been automatically scraped from web images. For the alignment task, the authors create unaligned images by randomly mismatching captions and images. For the masking task, 15% of the both the visual and language tokens are masked. The task is to reconstruct the mask from the remaining input in a classical Bert fashion. While the text masks are directly regressed like in Bert, the model predicts distributions over semantic classes for the image regions. This is achieved through minimizing the KL divergence, a measure for the similarity of distributions, between the output distribution of the pre-trained model used in feature extraction and the VilBert predictions.

The performance results are depicted in figure 3.44.

Table 1: Transfer task results for our ViLBERT model compared with existing state-of-the-art and sensible architectural ablations. [†] indicates models without pretraining on Conceptual Captions. For VCR and VQA which have private test sets, we report test results (in parentheses) only for our full model. Our full ViLBERT model outperforms task-specific state-of-the-art models across all tasks.

Method	VQA [3]		VCR [25]		RefCOCO+ [32]		Image Retrieval [26]		ZS Image Retrieval				
	test-dev (test-std)	Q→A	QA→R	Q→AR	val	testA	testB	R1	R5	R10	R1	R5	R10
SOTA	DFAF [36]	70.22 (70.34)	-	-	-	-	-	-	-	-	-	-	-
	R2C [25]	-	63.8 (65.1)	67.2 (67.3)	43.1 (44.0)	-	-	-	-	-	-	-	-
	MAttNet [33]	-	-	-	-	65.33	71.62	56.02	-	-	-	-	-
	SCAN [35]	-	-	-	-	-	-	48.60	77.70	85.20	-	-	-
Ours	Single-Stream [†]	65.90	68.15	68.89	47.27	65.64	72.02	56.04	-	-	-	-	-
	Single-Stream	68.85	71.09	73.93	52.73	69.21	75.32	61.02	-	-	-	-	-
	ViLBERT [†]	68.93	69.26	71.01	49.48	68.61	75.97	58.44	45.50	76.78	85.02	0.00	0.00
	ViLBERT	70.55 (70.92)	72.42 (73.3)	74.47 (74.6)	54.04 (54.8)	72.34	78.52	62.61	58.20	84.90	91.52	31.86	61.12
													72.80

FIGURE 3.44: Lu et al. (2019): VilBert Performance

As mentioned before, the dual stream architecture outperforms the single stream architecture. Furthermore, pretraining considerably boosts performance, as does finetuning. Interestingly, the authors also study the effect of the size of the dataset and effect of the architecture depth. Performance increases monotonically with dataset size, suggesting that performance can be further improved with more data. The results on the optimal layer depth are task dependent. VQA and Image Retrieval reach peak performance at 6 layers, where a layer denotes a repeatable block as depicted in figure 3.41. Zero Shot Image retrieval greatly benefits from even deeper depth. However, the VCR and RefCOCO+ tasks seemingly benefit from shallower models. The VQA task is based on the VQA 2.0 dataset. Each image must be matched to one of ten answers. Hence, the VQA task is not open-ended, but treated like a classification task. To achieve that, the model is amended by two MLP layers which use the elementwise product of the model-generated img and cls tokens. The VCR task is also posed as a multiple choice problem with images from movie scenes. To fine-tune for the task, questions and answers are concatenated into four different text input and given as model input together with the image. In the end, four scores are generated accordingly and selected through softmax. The RefCoCO+ task is a grounding task. An image region has to be selected according to a natural language reference. Caption-Based Image Retrieval requires the model to find an image that corresponds to a selected caption. The dataset used is the Flickr30k dataset which contains 30 000 pictures with five captions that are of higher quality than the automatically generated captions from web data.

3.5.3 Flamingo

The VilBert model showed one way how to actually combine visual and language inputs. In contrast, data2vec showed how to design an unsupervised

model and how influential the actual training process as well as contextualization can be. A natural question to ask is then is whether we can build a truly multimodal architecture like VilBert that is self-supervised like data2vec or at little task-specific training and how to optimized its training procedure. In particular, both VilBert and data2vec were tested on multiple tasks, but each task needs slight re-adjustments to the model as well as additional fine-tuning. Ideally, a multimodal architecture would not only be efficient in its initial training, but also easily adaptable to different tasks. Finding ways to not only work with different input modalities, but also with different task is crucial towards building a more general AI. A promising approach in that direction is few shot learning. The following section presents Flamingo (Alayrac et al. (XXXXa)), a few shot multimodal architecture developed by Google which comprises key innovations such as handling arbitrarily interleaved vislang sequences as inputs, as well as ways to effectively combine pretrained vision-only and language-only models. As such, it is a visually conditioned autoregressive text generation model.

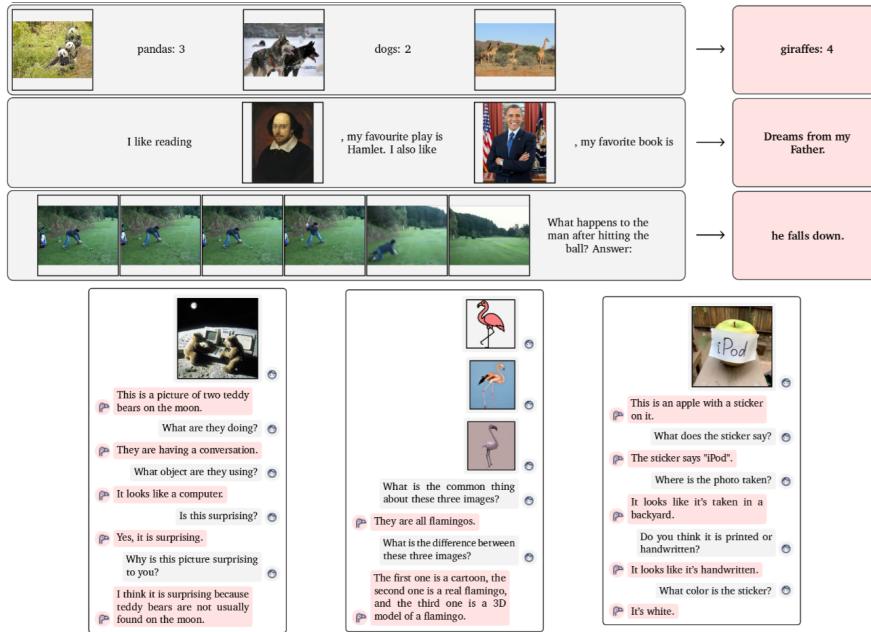


FIGURE 3.45: Alayrac et al. (2022): Flamingo Prompt-Output-Examples

Figure 3.45 demonstrates Flamingos capabilities. It can function as chat bot, describe pictures, work with image sequences (videos) and in doing so, simply needs a few prompts.

At the heart of the model is a large language model, Chinchilla (Hoffmann et al. (XXXX)), with 70B parameters. Large language models such as GPT-3

(Brown et al. (XXXXa)), as their name suggests, can be trained on a large amount of text data which gives them impressive text generative capabilities. However, multimodal generative modelling presents some specific challenges not present in language-only modelling. First of all, training large language models is expensive. Hence, it is paramount to work with a pretrained version, but trying to teach a large language model the means to work with visual inputs, as well, has the potential to deteriorate or destabilize the pretrained model. Second, large language models can suffer from memory constraints that are potentially severely aggravated by simply adding high-dimensional visual data into an input sequence. Third, good generalist capabilities typically require a huge amount of heterogeneous training data. There might not exist enough labelled image-caption-pair data to successfully accomplish training a capable few shot learning model in the vision-and-language domain. To train Flamingo, the authors solve these challenges by foremost exploring ways to generate their own web-scraped multimodal data set similar to existing ones in the language-only domain. Furthermore, they use a perceiver architecture (Jaegle et al. (XXXX)) that resamples inputs into a fixed amount of visual tokens. Finally, the self-attention layers of the language model are kept frozen during training while cross-attention layers are interleaved. A gating mechanism ensures that those new cross-attention layers do not interfere at model initialization, thereby improving stability and final performance.

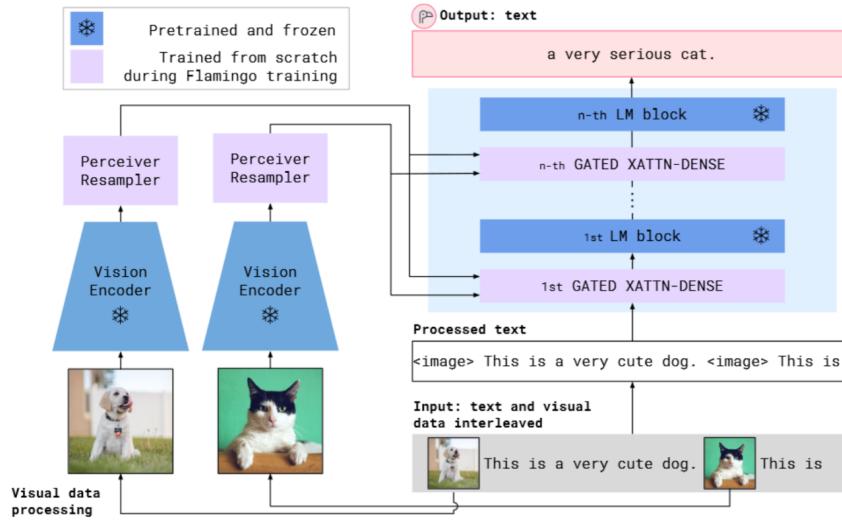


FIGURE 3.46: Alayrac et al. (2022): Flamingo Model Structure

Figure 3.46 shows the fundamental architecture of Flamingo. A pretrained vision model as well as a pretrained language model are frozen. Together they built the cornerstones of the model. The vision model is pretrained using a contrastive text-image approach. Its role is to extract features such as colour,

shape, nature and the position of objects - typical semantic spatial features that one would use in querying. The language model is an existing pretrained language model. On top of those frozen parts, the authors add a perceiver-resampler and gated cross-attention layers as learnable architectures. The perceiver-resampler turns the outputs of the vision model into a fix set of visual tokens. Those visual tokens are then used to create cross-attention layers which are interleaved into the frozen language model. As a result, Flamingo can model the likelihood of some text y interleaved with a sequence of images or videos x as

$$p(y|x) = \prod_{l=1}^L p(y_l|y_{<l}, x_{<l}).$$

Here, y_l denotes the language token associated with the input text and $(y, x)_{<l}$ is the set of preceding tokens. Parameterized by the model is p . As shown in the initial figure, one conditions the Flamingo model simply by giving it an alternating image and text sequence. This is because the attention mechanism only allows text tokens to attend to the previous image, which turned out to work better than alternative schemes. In particular, this means that the model can generalize to an arbitrary amount of images regardless of the amount of images used in training.

The training was based on multiple different datasets. The most important one is a Multimodal MassiveWeb datatset (M3M). To generate it, the authors collect images and text from HTML of approximately 43 million webpages. In the process, the position of images relative to the surrounding text is also identified. Image tags (image) added in plain text signal the original location of images to the model. In addition to that, end of chunk (EoC) tokens before images separate image-text sequences. The embedding of that token is added to the vocabulary of the language model with a random initialization that can later be learnt. Then, it is possible to infer the length of an image-text sequence, which is another piece of derived information. To give an idea of the scope of the dataset, M3M contains about 182GB of text as well as roughly 185 million images. The authors pay special attention not to include traditional task-specific datasets curated particularly for machine learning purposes to guarantee the generality of their modelling approach. As a second important dataset, aligned image text pairs are used. In particular, the ALIGN dataset ([Jia et al. \(XXXXb\)](#)).The dataset is further augmented with Long Text and Image Pairs (LTIP) as well as Video and Text Pairs (VTP). The later datasets contain more descriptive captions than ALIGN. Together the process ensures that the available training datasets are sufficiently large and heterogeneous - two key properties necessary to hopefully achieve good few shot performance.

The training objective is to minimize the weighted sum of dataset specific expected negative log likelihood.

$$\sum_{m=1}^M \lambda_m \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}_m} \left[- \sum_{\ell=1}^L \log p(y_\ell | y_{<\ell}, x_{\leq \ell}) \right]$$

Each dataset is weighted with a scalar λ as datasets can be of different quality or feature different properties. Hence, it might be preferable to pay different attention to different datasets. According to the authors, tuning these weights was essential to overall performance. In practice, the optimization works as follows: a sample batch with visual language sequences from each dataset is used to compute the gradient of the loss in accordance to the weight of the dataset. Importantly, the authors find that it is beneficial to accumulate the gradients of all datasets before triggering an updating process. Naturally, the actual datasets used to train the models are extremely crucial, as well. In their ablation studies, the authors find that removing their web-scraped multimodal dataset from the training pool drops model performance as measured across all selected tasks from a score of 68.4 to 46.9. Removing the dataset containing aligned captions and images drops performance to a score of 56.5 and not accumulating gradients before the updating process decreases performance to 59.7.

Taking a closer look at the model architecture, the two key structures are the perceiver resampler and the attention layers. Figure 3.47 shows the architecture of the perceiver. (Jaegle et al. (XXXX)) Before the input data reaches the perceiver, it is processed by the vision encoder - a Normalizer-Free ResNet which is trained with a contrastive loss similar to the well known Clip model (Radford et al. (XXXXa)) and yields a good trade-off between performance and efficiency. The output of the vision encoder is a 2D grid which is than flattened before being fed into the perceiver that connects the the vision encoder with the frozen language model. The resampling performed by the perceiver-resampler is crucial to reduce the complexity of vision-text cross-attention in the next step. This is particularly notable for video inputs. Inside the perceiver, a set of learned latent queries cross attend to the flattened vision encoder output. The number of outputs generated by the perceiver is equal to the number of learned latent queries. A change the authors make compared to previous work is to concatenate the keys and values from the latent queries with the keys and values from the flattened features. The ablation studies show that a medium sized perceiver architecture works best with improvements around two to three score points. Furthermore, a too large architecture can lead to instable trainings in conjunction with large frozen language model. The authors also test the perceiver against a transformer or MLP, which showed the perceiver to improve performance scores by around three to five points.

The main component that turns the frozen large language model into a functioning visual language model are the cross-attention layers depicted in figure 3.48. The number of layers added controls the number of free parameters and hence the complexity and expressiveness of the model. Keys and values of those

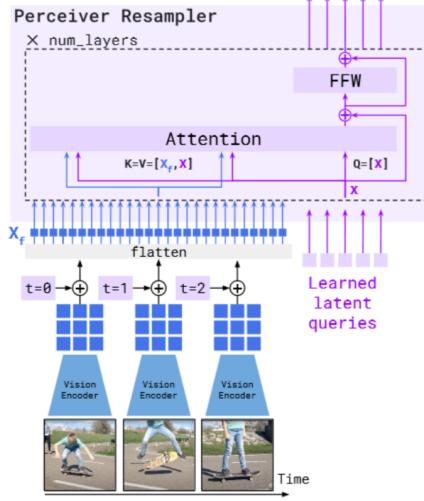


FIGURE 3.47: Alayrac et al. (2022): Flamingo Perceiver-Resampler

layers are obtained from the visual features output by the perceiver while using language queries. Specifically, gated cross-attention dense layers are used. The layers are dense because the cross-attention layer is followed by a feed forward layer. They are gated because a $\tanh(\alpha)$ gating mechanism is applied between layers. The gating mechanism ensures that the frozen large language model remains stable at initialization through introducing a learnable scalar parameter α initialized at 0. Without that initialization, training instabilities can occur. A key question is how many cross-attention layers should be used. For the small Flamingo model with 3B parameters, the ablation studies show that adding cross-attention between every self-attention layer of the frozen model yields the best results. However, adding further cross-attention layers does noticeably scale the parameter count of the model. A clear performance trade-off exists. After making hardware considerations, the authors settled for adding $\tanh(\alpha)$ gated cross-attention layers every 7th layer in the frozen large language model. The practical implementation of those attention layers works as follows: recall that the authors found that attending only to the nearest image improves performance by approximately 8 score points. To achieve this, while all text tokens attend to all visual tokens, a masking strategy is applied which ensures that in effect, language tokens only see a specific amount of visual tokens. Note however, that while the model can, unless specified otherwise, only attend to one image at a time, there is still a causal dependency to all previous images through the self-attention in the text-decoder.

To evaluate the model, the authors chose 18 different vison-language benchmarks including video benchmarks as shown in 3.49.

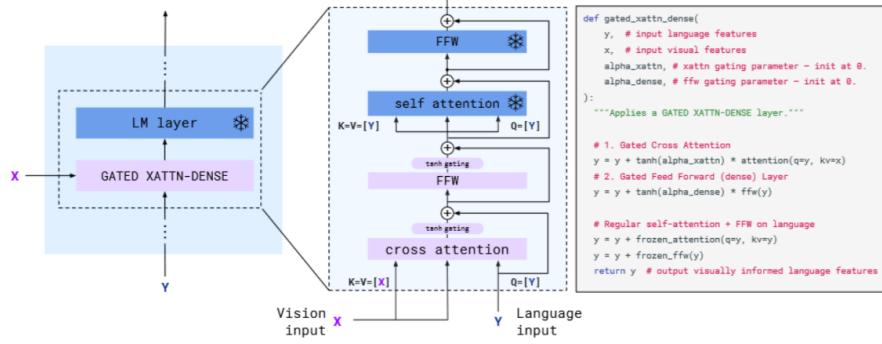


Figure 5 | **GATED XATTN-DENSE** layers. We insert new cross-attention layers, whose keys and values are obtained from the vision features while using language queries, followed by dense feed forward layers in between existing pretrained and frozen LM layers in order to condition the LM on visual inputs. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

FIGURE 3.48: Alayrac et al. (2022): Flamingo Gated Cross-Attention

	Dataset	DEV	Gen.	Custom prompt	Task description	Eval set	Metric
Image	ImageNet-1k [103]	✓			Object classification	Val	Top-1 acc.
	MS-COCO [18]	✓	✓		Scene description	Test	CIDEr
	VQAv2 [3]	✓	✓		Scene understanding QA	Test-dev	VQA acc. [3]
	OKVQA [75]	✓	✓		External knowledge QA	Val	VQA acc. [3]
	Flickr30k [149]		✓		Scene description	Test (Karpathy)	CIDEr
	VizWiz [40]		✓		Scene understanding QA	Test-dev	VQA acc. [3]
	TextVQA [108]		✓		Text reading QA	Val	VQA acc. [3]
	VisDial [22]				Visual Dialogue	Val	NDCG
	HatefulMemes [60]			✓	Meme classification	Seen Test	ROC AUC
Video	Kinetics700 2020 [110]	✓			Action classification	Val	Top-1/5 avg
	VATEX [132]	✓	✓		Event description	Test	CIDEr
	MSVDQA [140]	✓	✓		Event understanding QA	Test	Top-1 acc.
	YouCook2 [161]		✓		Event description	Val	CIDEr
	MSRVTQA [140]	✓			Event understanding QA	Test	Top-1 acc.
	iVQA [145]		✓		Event understanding QA	Test	iVQA acc. [145]
	RareAct [81]			✓	Composite action retrieval	Test	mWAP
	NextQA [139]	✓			Temporal/Causal QA	Test	WUPS
	STAR [138]				Multiple-choice QA	Test	Top-1 acc.

FIGURE 3.49: Alayrac et al. (2022): Flamingo Datasets (Table2, p.19)

Note that seven benchmarks are used to validate design choices of the architecture. They are part of the development (dev) set. As those datasets could potentially report biased performance results, the remaining eleven datasets are solely used to estimate unbiased performance scores. Unfortunately, unbiased estimation in few-shot learning is not ubiquitous. Since hyperparameter tuning requires more prompts, it is easy to forget about them when counting how many shots in effect have been used, which can in turn lead to overestimation of performance (Perez et al. (XXXX)). As far as the Flamingo model is concerned, the authors take great care to evaluate it in a true few-shot fashion as coined by Perez et al. (XXXX). Furthermore, most of the tasks require

a generative answer (gen) which encompasses open-ended, more interesting tasks.

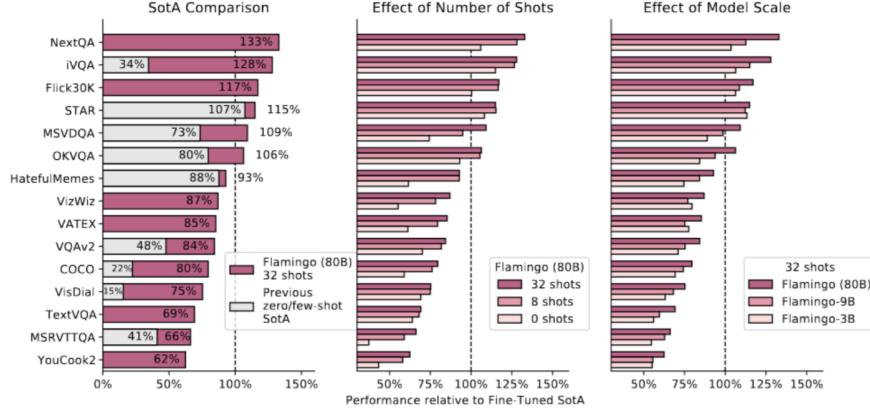


FIGURE 3.50: Alayrac et al. (2022): Flamingo Results without Fine-Tuning

The results portayed in figure 3.50 show that Flamingo does not only outperform all previous few shot models, but also the general state-of-the art on six tasks. It is also, not too surprisingly, evident that more shots and larger models lead to better performance. The in-context learning works analogous to GPT-3 (Brown et al. (XXXXa)). Given a set of supporting examples (image, text), where text is the expected response based on the supporting visual input, a multimodal prompt is built by concatenating the examples in random order and adding the selected query image for the prediction. Interestingly, rather than using in-context learning with prompts, the authors also explore fine-tuning the model on the tasks which achieved state-of-the-art performance with few-shot learning. Fine-tuning the model is very expensive and requires additional hyperparameter tuning, but substantially improves results even further.

One noticeable exception that the authors remark upon is the classification performance of the model. In that realm, contrastive models outperform Flamingo. A working hypothesis is that training for text-image retrieval is particularly important on those tasks. Nevertheless, few-shot learning and open-ended generative capabilities provide great advantages over contrastive models both in terms of model training as well as the range of applications. Flamingo shows how to leverage a unifying stucture between models. Communication between models is key to reducing training overload and to enable multitasking models.

Method	VQAv2		COCO		VATEX		VizWiz		MSRVTTQA		VisDial		YouCook2		TextVQA		HatefulMemes	
	test-dev	test-std	test	test	test-dev	test-std	test	valid	test-std	valid	valid	test-std	valid	valid	test-std	test seen		
Flamingo - 32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	-	-	-	70.0		
SimVLM [134]	80.0	80.3	143.3	-	-	-	-	-	-	-	-	-	-	-	-	-		
OFA [129]	79.9	80.0	<u>149.6</u>	-	-	-	-	-	-	-	-	-	-	-	-	-		
Florence [150]	80.2	80.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Flamingo Fine-tuned	82.0	82.1	138.1	84.2	65.7	65.4	47.4	61.8	59.7	118.6	57.1	54.1	86.6	-	-	-		
Restricted SotA [†]	80.2	80.4	<u>143.3</u>	76.3	-	-	46.8	75.2	74.5	138.7	54.7	73.7	75.4	[142]	[92]	[60]		
Unrestricted SotA	81.3	81.3	149.6	81.4	57.2	60.6	-	-	75.4	-	-	-	84.6	-	-	[164]		
	[143]	[143]	[129]	[165]	[70]	[70]	-	-	[133]	-	-	-	-	-	-	-		

Table 6 | Comparison to SotA when fine-tuning Flamingo. We fine-tune Flamingo on all nine tasks where Flamingo was SotA overall with few-shot learning. Flamingo sets a new SotA on five of these tasks sometimes even beating methods that resort to known performance optimization tricks such as model ensembling (on VQAv2, VATEX, VizWiz and HatefulMemes). Best numbers among the restricted SotA are in **bold**. Best numbers overall are underlined. Restricted SotA[†]: only includes methods that use a single model (not ensembles) and do not directly optimise the test metric (no CIDEr optimisation).

FIGURE 3.51: Alayrac et al. (2022): Flamingo Results with Fine-Tuning

3.5.4 Discussion

In the previous sections, we've analyzed three different models with a common goal, to work towards a more general model capable of succeeding on multiple tasks across multiple modalities, specifically image and text. Which lessons can be learned from that?

Context is key. Across all three models, it became apparent that **larger models have an edge**. A clever architecture can of course lead to great results - after all Flamingo outperformed many finetuned models, but ceteris paribus, larger models will deliver stronger performance. While Flamingo with 80B parameters, mostly made up by its large language model component, was far larger than data2vec or VilBert, it is also far from the largest model in existence, as the following chapter will show. More tasks, more generalizability also means larger models.

Larger models in turn require either incredible resources or clever designs to be trained - likely both. It is not without reason that the three models presented have been developed by Meta and Google. At the high end, the amount of people with access to the necessary resources is naturally limited and there is little reason to expect change. On top of resources constraints limiting access to mostly a few private companies, resulting models are often also simply not publicly available to shield intellectual property from competitors. For example, while the codebase for the somewhat older VilBert model is publicly available, data2vec and Flamingo are not generally accessible.

At any rate, even for the companies with the greatest resources, **a performance-cost trade-off exists**. The question of how to cut down on required training time is essential. The general approach is to pretrain models and then fine-tune them on specific tasks. However, **few shot in-context**

learning provides a resource friendly alternative although fine-tuning likely still leads to better absolute results. **Freezing models**, particularly large language models, is a key idea on top of pretraining. In some cases, it is paramount to avoid the loss of capabilities that can go along with retraining a model. This could already be seen when VilBerts dual-stream architecture outperformed a single-stream design, but becomes more notable in the Flamingo architecture, where retaining the full expressiveness of the large language model is key which prompted the authors to introduce a gating mechanism into the cross-attention layers to stabilize the training process. In general, model collapse is always a concern, in particular when working with latent representations such as data2vec. In essence, rather than building single models from scratch, reusing models and leveraging communication between models is a new, promising approach. In that regard, **Socratic models** (Zeng et al. (XXXX)) also show that the knowledge stored in different models is symbiotic which they used for exciting tasks such as multimodal assisted dialogue with people or robot perception. Finally, **data matters**. Not only is the amount of data important, but also its composition. Heterogenous data are important, but so is the optimization across datasets. The Flamingo model was specifically trained with a weighted loss across datasets and it was possible to quantify the performance contribution of each of them. Particularly in few shot learning settings, it is thereby important to be careful about unbiased performance estimation as Perez et al. (XXXX) noted. Otherwise, it is easy to overestimate performance.

In any case, the quest towards more general, more unified models is far from completed. The common theme is to combine larger and larger models while employing resource friendly training regimes. For example **Pathway** models (Chowdhery et al. (XXXXa)), which will be further discussed in the upcoming chapter, use sparse activation which reduces the energy consumption to less than 10% of what would be expected from similar dense models.

4

Further Topics

Authors: Marco Moldovan, Rickmer Schulte, Philipp Koch

Supervisor: Rasmus Hvingelby

So far we have learned about multimodal models for text and 2D images. Text and images can be seen as merely snapshots of the sensory stimulus that we humans perceive constantly. If we view the research field of multimodal deep learning as a means to approach human-level capabilities of perceiving and processing real-world signals then we have to consider lots of other modalities in a trainable model other than textual representation of language or static images. Besides introducing further modalities that are frequently encountered in multi-modal deep learning, the following chapter will also aim to bridge the gap between the two fundamental sources of data, namely structured and unstructured data. Investigating modeling approaches from both classical statistics and more recent deep learning we will examine the strengths and weaknesses of those and will discover that a combination of both may be a promising path for future research. Going from multi modalities to multi task, the last section will then broaden our view of multi-modal deep learning by examining multi-purpose modals. Discussing cutting-edge research topics such as the newly proposed Pathways, we will discuss current achievements and limitations of the new modeling that might lead our way towards the ultimate goal of AGI in multi-modal deep learning.

4.1 Including Further Modalities

Author: Marco Moldovan

Supervisor: Rasmus Hvingelby

4.1.1 Intro

In this chapter we will build up a taxonomy of different perceivable and interpretable types of signals that we as humans use to navigate the world and we

will see how today's state-of-the-art multimodal models are built and trained in order to process more and more modalities simultaneously in order to build more and more complete representations of world through available data. We will build up our taxonomy starting from the two most well-understood modalities - namely text and 2D images - and introduce models that learn relationships between increasingly many modalities at the same time and to map them to a cross-modal representation space in which we can apply distance functions to points in order to represent semantic relatedness between datapoint from these different modalities. Given such a learned cross-modal representation space we will look at some of the most important multimodal downstream tasks and applications.

Towards the end of the chapter we will take a closer look at the two main types of model architectures and training paradigms: bi-encoders and "true" multimodal cross-encoders. The first kind of model can be seen as an ensemble of unimodal expert models that map into the same representation space while using some form of metric learning to relate representations of different modalities to one another. True multimodal models are essentially agnostic to their input (as long as it is preprocessed and featurized appropriately). We currently see the second kind of architecture as the more promising one for the case of approximating human-level perception. An example of a modality agnostic multimodal model is the Perceiver of which we will introduce a newer, even more efficient variant.

Up until recently each modality required their own specific self-supervised training paradigm: for text a common approach would be MLM while the same training paradigm wasn't as effective for images or video. data2vec introduces a modality-agnostic SSL masked prediction setup which requires careful preprocessing but does not care about the source of the input. We see a model that marries a modality-agnostic model like Perceiver with a modality agnostic training paradigm like data2vec as a very promising path forward. Topic 11 will build on this idea of modality-agnostic models by introducing Google's Pathways: a concept for multimodal, multi-task, sparse world models.

4.1.2 Motivation

- World is inherently multimodal, images and text are just discrete snapshots while we as humans perceive lots of continuous multimodal signals.
- We can extend the ideas and intuitions of image-text multimodal learning to include more modalities.
- Listing research that includes continuously more modalities would get out of hand quickly and seems unstructured: If we were to consider all possible learnable permutations of signal types we could go on forever.

4.1.3 Taxonomy of Multimodal Challenges

- Instead we want to build a taxonomy for multimodal machine learning that is based on challenges instead of modalities.
- Viewing multimodal learning from the perspective of challenges is more generic and intuitive.
- Once we understand the challenges we will see that real-world problems and their solutions will arise naturally.
- The taxonomy will act as a blueprint for approaching multimodal learning challenges. For each category we will introduce some examples that apply a mixture of diverse modalities to a model.
- We hope that the reader will understand that the different modalities are in principle interchangable and that he/she will be able to apply the correct framework to their own multimodal problem.

4.1.3.1 Multimodal Representation Learning

- Representation learning lies at the base of solving most learning problems today, including for multimodal learning problems.
- Dense representations are commonly learnt by deep neural networks like we've seen in the previous chapters.
- If we want to generalize this notion to an arbitrary number of modalities we have to be clear about the type of function that we want to learn and the kind of Representation we want to project to.

4.1.3.1.1 Joint Representations

- Different modalities live in the same representation space.
- Given a multimodal signal one needs to learn a model that "fuses" these modalities in order to learn a joint representation of these input signals.
- Typically modalities are somehow concatenated as an input and are then fed into a model that is constructed such as to learn a joint representation.

4.1.3.1.2 Coordinated Representation

- Given input signals of different modalities we can learn a class of models that each projects a single modality into its own space.
- One model will typically receive one modality.
- For learning joint representations we have to define a training paradigm that will learn to coordinate these different representation spaces by placing semantically similar representations close to each other while maximizing the distance between semantically different representations.
- Essentially one learns to align different representation spaces to each other.
- Contrastive learning is a popular paradigm for learning joint representations.

4.1.3.2 Multimodal Translation

- Given a signal in one modality we want to return a semantically equivalent output in a different modality.
- E.g. give a text input and retrieve a speech segment.
- E.g. provide a video and return a text description of the video
- Translation can be retrieval-based or generative. I.e. either return an existing datapoint or sample and synthesize a new one.
- Clip is classic example (already known)
- DALL-E is generative translation model
- NÜWA even more general across modalities
- VideoCLIP for video retrieval
- SpeechBERT for text to speech retrieval

4.1.3.3 Multimodal Alignment

- Multimodal alignment is the challenge of how exactly to learn coordinated representation spaces.
- VATT learns coordinated space contrastively. Can even share weights between modalities to serve as one-model-fits-all for multimodal alignment.
- Alternative: masked multimodal autoencoding with MultiMAE.

4.1.3.4 Multimodal Fusion

- Multimodal fusion is the challenge of how to learn a joint representation space.
- Where in the model does fusion happen? Early, late or hybrid fusion possible. What are the advantages and disadvantages of each approach?
- Introduce Multimodal Bottleneck Transformer (MBT)

4.1.4 General Multimodal Architectures

- Introduce architectures that are general enough to be applied to most/any multimodal problem
- NÜWA 3D Nearby Attention can take text, audio, images and video as input: data has to first be encoded into this format batch size x time x height x width x embedding size. Not necessarily suitable for joint representations but can serve as a general modality-agnostic encoder for coordinated representations. Time dimension can be replaced with channel or depth dimensions if one wants to encode more exotic modalities. Preserves locality in data.
- Perceiver and Perceiver IO require almost no preprocessing and can read ex-

tremely long sequences of data by cross-attending between data and modality specific learnable latent array.

- Hierarchical Perceiver is follow-up that can preserve locality and compositionality in data (very important).

4.1.5 Multimodal Training Paradigms

- Present training paradigms of how to train multimodal modal with SSL.

4.1.5.1 Modality-Agnostic Uni-Modal SSL

- data2vec is unimodal SSL paradigm but it's completely modality agnostic.

4.1.5.2 Generalized Cross-Modal SSL

- Here we introduce methods for true multimodal SSL.
- Have to separate into contrastive and non-contrastive methods
- Generalize as much as possible: are there any approaches that solve alignment and fusion at the same time?

4.1.5.2.1 Contrastive Methods

- VATT -> look for similar

4.1.5.2.2 Non-Contrastive Methods

- MultiMAE -> look for similar
- Can data2vec work with multiple modalities?

4.1.6 Combining General Architectures and Training Paradigms

- Future research: combining general architectures like Perceiver with contrastive methods like VATT, data2vec or MultiMAE.

4.2 Structured + Unstructured Data

Author: Rickmer Schulte

Supervisor: Daniel Schalk

4.2.1 Intro

While the previous chapter has extended the range of modalities considered in multimodal deep learning beyond image and text data, the focus remained on other sorts of unstructured data. This has neglected the broad class of structured data, which has been the basis for research in pre-deep learning eras and which has given rise to many fundamental modeling approaches in statistics and classical machine learning. Hence, the following chapter aims to give an overview of both data sources and will outline the respective ways how these have been used for modeling purposes as well as more recent attempts to model them jointly.

Generally, structured and unstructured data substantially differ in certain aspects such as dimensionality and interpretability. This has led to various modeling approaches that are particularly designed for the special characteristics of the data types, respectively. As shown in previous chapters, deep learning models such as neural networks are known to work well on unstructured data. This is due to their ability to extract latent representation and to learn complex dependencies from unstructured data sources to achieve state-of-the art performance on many classification and prediction tasks. By contrast, classical statistical models are mostly applied on tabular data due the advantage of interpretability inherent to these models, which is commonly of great interest in many research fields. However, as more and more data has become available to researchers today, they often do not only have one sort of data modality at hand but both structured and unstructured data at the same time. Discarding one or the other data modality makes it likely to miss out on valuable insights and potential performance improvements.

Therefore, in the following sections we will investigate different proposed methods to model both data types jointly and examine similarities and differences between those. Different fusion strategies to integrate both types of modalities into common deep learning architectures are analyzed and evaluated, thereby touching upon the concept of end-to-end learning and its advantages compared to separated multi-step procedures. The different methods will be explored in detail by referring to numerous examples from survival analysis, finance and economics. Finally, the chapter will conclude with a critical assessment of recent research for combining structured and unstructured data in multimodal DL, highlighting limitations and weaknesses of past research as well as giving an outlook on future developments in the field.

4.2.2 Taxonomy: Structured vs. Unstructured Data

In order to have a clear setup for the remaining chapter, we will start off with a brief taxonomy of data types that will be encountered. Structured data, normally stored in a tabular form, has been the main research object in classical scientific fields. Whenever there was unstructured data involved, this was normally transformed into structured data in an informed manner. Typically, doing so by applying expert knowledge or data reduction techniques such as PCA prior to further statistical analysis. However, DL has enabled unsupervised feature extraction from unstructured data and thus to feed it to the models directly. Classical examples of unstructured data are image, text, video, and audio data as shown in the figure below. Of these, image data in combination with tabular data is the most frequently encountered. Hence, this combination will be examined along various examples later in the chapter. While previously mentioned data types allow for a clear distinction, lines can become increasingly blurred. For example, the record of a few selected biomarkers or genes from patients would be regarded as structured data and normally be analyzed with classical statistical models. On the contrary, having the records of multiple thousand biomarkers or genes would rather be regarded as unstructured data and usually be analyzed using DL techniques. Thus, the distinction between structured and unstructured data does not only follow along the line of dimensionality but also concerns the interpretability of single features within the data.

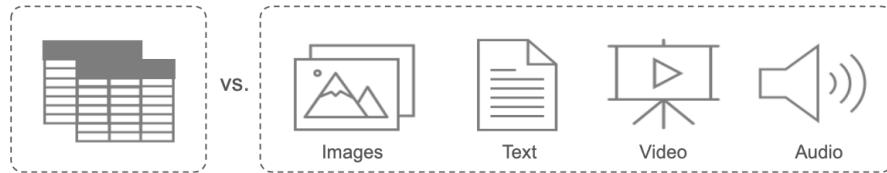


FIGURE 4.1: Structured vs. Unstructured Data

4.2.3 Fusion Strategies

After we have classified the different data types that we will be dealing with, we will now discuss different fusion strategies that are used to merge data modalities into a single model. While there are potentially many ways to fuse data modalities, a distinction between three different strategies, namely early, joint and late fusion has been made in the literature. Here we follow along the taxonomy laid out by [Huang et al. \(2020\)](#) with a few generalizations as those are sufficient in our context.

Early fusion refers to the procedure of merging data modalities into a common feature vector already at the input layer. The data that is being fused

can be raw or preprocessed. The step of preprocessing usually involves dimensionality reduction to align dimensions of the input data. This can be done by either training a separate DNN (Deep Neural Network), using data driven transformations such as PCA or directly via expert knowledge.

Joint fusion offers the flexibility to merge the modalities at different depths of the model and thereby to learn latent feature representations from the input data (within the model) before fusing the different modalities into a common layer. Thus, the key difference to early fusion is that latent feature representation learning is not separated from the subsequent model. This allows backpropagation of the loss to guide the process of feature extraction from raw data. The process is also called end-to-end learning. Depending on the task, CNNs or LSTMs are usually utilized to learn latent feature representations. As depicted in the figure below, it is not required to learn lower dimensional feature representations for all modalities and is often only done for unstructured data. A further distinction between models can be made regarding their model head, which can be a FCNN (Fully Connected Neural Network) or a classical statistical model (linear, logistic, GAM). While the former can be desirable to capture possible interactions between modalities, the latter is still frequently used as it preserves interpretability.

Late fusion or sometimes also called decision level fusion is the procedure of fusing the predictions of multiple models that have been trained on each data modality separately. The idea originates from ensemble classifiers, where each model is assumed to inform the final prediction separately. Outcomes from the models can be aggregated in various ways such as averaging or majority voting.

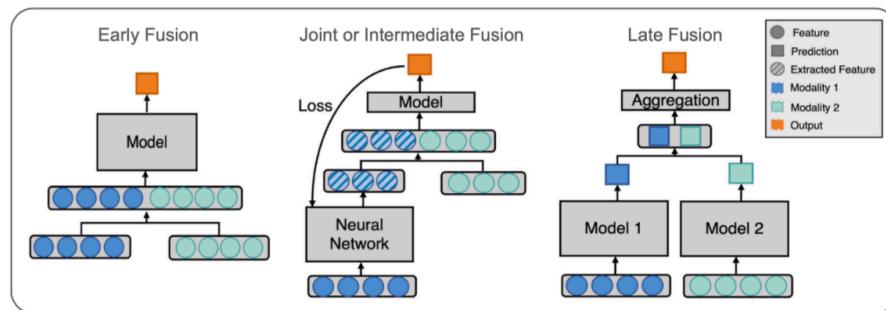


FIGURE 4.2: Data Modality Fusion Strategies (Adopted from Huang et al., 2020).

We will refer to numerous examples of both early and joint fusion in the following sections. While the former two are frequently applied and easily comparable, late fusion is less common and different in nature and thus not further investigated here. As a general note, for the sake of simplicity we will refer to the special kind of multimodal DL including both structured and

unstructured data when we speak about multimodal DL in the rest of the chapter.

4.2.4 Applications

The following section will discuss various examples of this kind of multimodal DL by referring to different publications and their proposed methods. The publications originate from very different scientific fields, which is why methods are targeted for their respective use case. Hence, allowing the reader to follow along the development of methods as well as the progress in the field. Thereby, obtaining a good overview of current and potential areas of applications. As there are various publications related to this kind of multimodal DL, the investigation is narrowed down to publications which either introduce new methodical approaches or did pioneering work in their field by applying multimodal DL.

4.2.4.1 Multimodal DL in Survival

Especially in the field of survival analysis, many interesting ideas were proposed with regards to multimodal DL. While clinical patient data such as electronic health records (EHR) were traditionally used for modeling hazard functions in survival analysis, recent research has started to incorporate image data such as CT scans and other modalities such as gene expression data in the modeling framework. Before examining these procedures in detail, we will briefly revisit the classical modeling setup of survival analysis by discussing the well-known Cox Proportional Hazard Model (CPH).

4.2.4.2 Traditional Survival Analysis (CPH Model)

Survival Analysis generally studies the time duration until a certain event occurs. While many methods have been developed to analyze the effect of certain variables on the survival time, the Cox Proportional Hazard Model (CPH) remains the most prominent one. The CPH models the hazard rate which is the conditional probability of a certain event occurring in the next moment given that it has not so far:

$$h(t|x) = h_0(t) * e^{x\beta}$$

where $h_0(t)$ denotes the baseline hazard rate and β the linear effects of the covariates x on which the probability is conditioned on. The fundamental assumption underlying the traditional CPH is that covariates influence the hazard rate proportionally and multiplicatively. This stems from the fact that the effects in the so-called risk function $f(x) = x\beta$ are assumed to be linear. Although this has the advantage of being easily interpretable, it does limit the

flexibility of the model and thus also the ability to capture the full dynamics at hand.

4.2.4.3 Multimodal DL Survival Analysis

Overcoming the limitations of the classical CPH model, Katzman et al. (2018) were among the first to incorporate neural networks into the CPH and thereby replacing the linear effect assumption. While their so-called DeepSurv model helped to capture interactions and non-linearities of covariates, it only allowed modeling of structured data. This gave rise to the model DeepConvSurv of Zhu et al. (2016), who apply CNNs to extract information from pathological images in order to predict risk of patients subsequently. They showed that learning features from images via CNNs in an end-to-end fashion outperforms methods that relied on hand-crafting features from these images. Building on the idea of DeepConvSurv, ? extended the model by adding further modalities. Besides pathological images, their proposed DeepCorrSurv model also includes molecular data of cancer patients. The name of the model stems from the fact that separate subnetworks are applied to each modality and that the correlation between the output of these modality specific subnetworks are maximized before fine-tuning the learned feature embedding to perform well on the survival task. The correlation maximization procedure aims to remove the discrepancy between modalities. It is argued that the procedure is beneficial in small sample settings as it may reduce the impact of noise inherent to a single modality that is unrelated to the survival prediction task.

The general idea is that the different modalities of multimodal data may contain both complementary information contributed by individual modalities as well as common information shared by all modalities. The idea was further explored by subsequent research. Tong et al. (2018) for example introduced the usage of auto encoders (AE) in this context by proposing models that extract the lower dimensional hidden features of the AE applied to each modality. While their first model trains AEs on each modality separately before concatenating the learned features (ConcatAE), their second model obtains cross-modality AEs that are trained to recover both modalities from each modality respectively (CrossAE). Here, the concept of complementary information of modalities informing survival prediction separately gives rise to the first model, whereas the concept of retrieving common information inherent across modalities gives rise to the latter. Although, theoretically both models could also handle classical tabular EHR data, they were only applied to multi-omics data such as gene expressions of breast cancer patients.

Similar to Tong et al. (2018), Cheerla and Gevaert (2019) also derive their model from the idea of common information that is shared by all modalities. Besides, having specialized subnetworks for each modality to learn latent feature embeddings, they also introduce a similarity loss that is added to the classical cox loss from the survival prediction. This similarity loss is applied

to each subnetwork output and aims to learn modality invariant latent feature embeddings. This is desirable not only for noise reduction but also in cases of missing data. While previous research often applied their models only on subsets of the large cancer genome atlas program (TCGA), [Cheerla and Gevaert \(2019\)](#) analyze 20 different cancer types of the TCGA using four different data modalities. As expanding the scope of the study increases the problem of data missingness, they specifically target the problem by introducing a variation of regular dropout, which they refer to as multimodal dropout. Instead of dropping certain nodes, multimodal dropout drops entire modalities during training in order to make models less dependent on one single data source. This enables the model to better cope with missing data during inference time. Opposed to [Tong et al. \(2018\)](#), the model is trained in an end-to-end manner and thus allows latent feature learning to be guided by the survival prediction loss. More impressive than their overall prediction performances are the results of T-SNE-mappings that are obtained from the learned latent feature embeddings. One sample mapping is displayed in the figure below, which nicely shows the clustering of patients with regards to cancer types. This is particularly interesting regarding the fact that the model was not trained on this variable. Besides being useful for accurate survival prediction, such feature mappings can directly be used for patient profiling and are thus pointed out as a contribution to the research on their own.

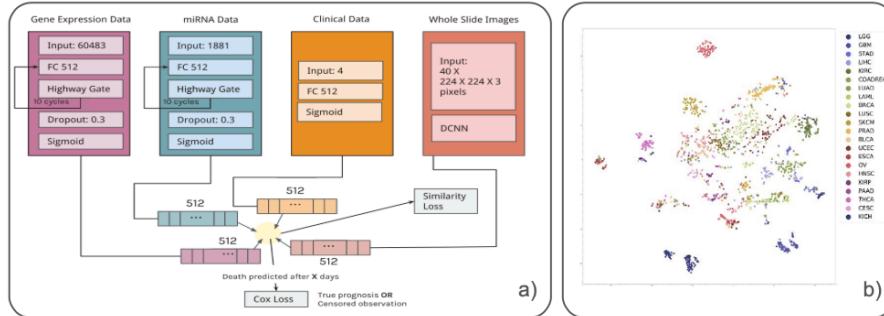


FIGURE 4.3: a) Architecture with Similarity Loss b) T-SNE-Mapped Representations of Latent Features (Colored by Cancer Type) ([Cheerla and Gevaert, 2019](#)).

[Vale-Silva and Rohr \(2021\)](#) extend the previous work by enlarging the scope of study, analyzing up to six different data modalities and 33 cancer types of the TCGA dataset. Their so-called MultiSurv model obtains a straightforward architecture, applying separate subnetworks to each modality and a subsequent FCNN (model head) to yield the final survival prediction. Testing their modular model on different combinations of the six data modalities, they find the best model performance for the combination of structured clinical and mRNA data. Interestingly, including further modalities lead to

slight performance reductions. Conducting some benchmarking, they provide evidence for their best performing model (structured clinical + mRNA) to outperform all single modality models. However, it is worthwhile mentioning that their largest model, including all six modalities, is not able to beat the classical CPH model, which is based on structured clinical data only. While this already may raise concerns about the usefulness of including so many modalities in the study, high variability of model performance between the 33 cancer types is also found by the authors and may indicate a serious data issue. The finding may seem less surprising, considering the fact that tissue appearances can differ vastly between cancer types. This is particularly problematic as for some of these cancer types only very few samples were present in the training data. For some there were only about 20 observations in the training data. Although state-of-the-art performance is claimed by the authors, the previously mentioned aspects do raise concerns about the robustness of their results. Besides, facing serious data quantity issues for some cancer types, results could simply be driven by the setup of their analysis by testing the model repeatedly on different combinations of data modalities. Thereby increasing the chances to achieve better results at least for some combinations of data modalities. Moreover, the study nicely showcases that the most relevant information can often be retrieved from classical structured clinical data and that including further modalities can by contrast even distort model training when sample sizes are low compared to the variability within the data. While these concerns could certainly have been raised for the other studies as well, they simply become more apparent in [Vale-Silva and Rohr \(2021\)](#) due their comprehensive and transparent analysis.

In the last part of this section we will refer to a different set of survival models by introducing the concept of Wide & Deep NN. The idea for Wide & Deep NN was first introduced by [Cheng et al. \(2016\)](#), who proposed to not only feed data inputs to either a linear or FCNN model part, but both at the same time. Applying it in the context of Recommender Systems, the initial assumption was that models need to be able to memorize as well as generalize for prediction tasks and that these aspects could be handled by the linear and FCNN part, respectively.

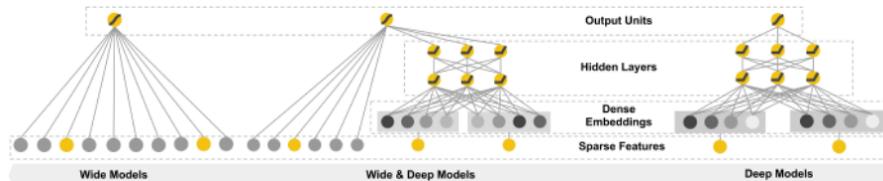


FIGURE 4.4: Illustration of Wide & Deep Neural Networks ([Cheng et al., 2016](#)).

The idea of Wide & Deep NN is applied in the context of multimodal DL

survival by Pölsterl et al. (2019) and Kopper et al. (2022). Similar to previous studies Pölsterl et al. (2019) make use of the CPH model and integrate Wide & Deep NN in these. By contrast, Kopper et al. (2022) integrate them in a different set of survival models, namely the piecewise exponential additive mixed model (PAMM). Without going into detail, the general purpose of this model class is not only to overcome the linearity but also the proportionality constraint in the classical CPH. By dropping the proportionality assumption, these models yield piecewise constant hazard rates for predetermined time intervals. Although the two studies differ in their model setup, both studies leverage structured as well as visual data and additionally make use of a linear model head. The latter is particularly interesting as it is this additive structure in the last layer of the models which preserves interpretability. Thus, the obtain models that not only have the flexibility for accurate predictions itself but which are also able to recover the contributions of single variables to these predictions.

Although, Wide & Deep NN are advantageous due to their flexibility and interpretability, special care needs to be taken regarding a possible feature overlap between the linear and NN part as it can lead to an identifiability problem. This can be illustrated by considering the case that a certain feature x is fed to the linear as well as the FCNN model part. Because of the Universal Approximation Theorem for Neural Networks, it is known that the FCNN part could potentially model any arbitrary relation between the dependent and independent variable ($d(x)$). However, this is what raises the identifiability issue as the coefficients (β) of the linear part could theoretically be altered arbitrarily ($\tilde{\beta}$) without changing the overall prediction when the weights of the NN ($\tilde{d}(x)$) are adjusted accordingly.

$$x\beta + d(x) = x\tilde{\beta} + \tilde{d}(x) + f(x) = x\tilde{\beta} + \tilde{d}(x)$$

Generally, there are two ways to deal with this identifiability problem. The first possibility would be to apply a two-stage procedure by first estimating only the linear effects and then applying the DL model part only on the obtained residuals. An alternative would be to incorporate orthogonalization within the model, thereby performing the procedure in one step and allowing for efficient end-to-end training. The latter was proposed by Rügamer et al. (2020) and utilized in the DeepPAMM model by Kopper et al. (2022). The next section will go into more detail about the two possibilities to solve the described identifiability issue and proceed by discussing further applications of multimodal DL in other scientific fields.

4.2.4.4 Multimodal DL in Other Scientific Fields

After having seen multiple applications of multimodal DL in survival analysis which predominantly occurs in the biomedical context, we will now extend

the scope of the chapter by discussing further applications of multimodal DL related to the field of economics and finance. While structured data has traditionally been the main studied data source in these fields, recent research has not only focused on combining both structured and unstructured data, but also on ways to replace costly collected and sometimes scarcely available structured data with freely available and up-to-date unstructured data sources using remote sensing data. Before examining these approaches, we will first go into more detail about the model proposed by Rügamer et al. (2020), which not only introduced a new model class in the context of multimodal DL but also offered a method to efficiently solve the above mentioned identifiability problem.

As previous research exclusively focused on mean prediction, uncertainty quantification has often received less attention. Rügamer et al. (2020) approach this by extending structured additive distributional regression (SADR) to the DL context. Instead of learning a single parameter e.g. the mean, SADR provides the flexibility to directly learn multiple distributional parameters and thereby natively includes uncertainty quantification. It is nevertheless possible to only model the mean of the distribution, which is why SADR can be regarded as a generalization of classical mean prediction. Rügamer et al. (2020) now extend this model class by introducing a framework that can model these distributional parameters as a function of covariates via a linear, generalized additive (GAM) or NN model. All distributional parameters are resembled in a final distributional layer (output layer). An illustration of their so-called Semi-Structured Deep Distributional Regression (SSDDR) is given in the figure below.

If the mean is now modeled by both a linear and DNN part and the same feature inputs are fed to both model parts, we are in the setting of Wide & Deep NN. As illustrated above, such feature overlaps give rise to an identifiability issue. The key idea of Rügamer et al. (2020) to mitigate this problem was to integrate an orthogonalization cell in the model, that orthogonalizes the latent features of the deep network part with respect to the coefficients of the linear and GAM part if feature overlaps are present. More precise, in case X contains the inputs, that are part of the feature overlap, the projection matrix \mathcal{P}^\perp projects into the respective orthogonal complement of the linear projection which is on the column space spanned by X . This allows backpropagation of the loss through the orthogonalization cell and therefore enables end-to-end learning. As the linear and GAM effect channels are directly connected to the distributional layer, the orthogonalization cell is therefore able to preserve the interpretability of the model.

Another way of orthogonalizing feature representations is by applying a two-stage procedure as described above. ? utilize this procedure to make their latent feature representations retrieved from unstructured data orthogonal to their linear effect estimates from structured data. More specifically, they try to accurately predict house prices in London using multimodal DL on street

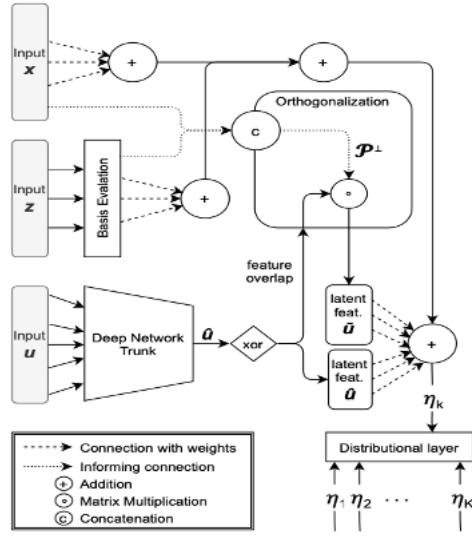


FIGURE 4.5: Architecture of SSDDR (X+Z (Struct.) and U (Unstruct.) Data) (Rügamer et al., 2020).

and aerial view images as well as tabular housing attributes. Applying the two-stage procedure they aim at learning latent feature representations from the image data which only incorporate features that are orthogonal to the housing attributes. Thereby, they limit the chances of confounding in order to obtain interpretable housing attribute effects. Conducting a series of experiments, they find that including image data next to the tabular housing data does improve the prediction performance over single modality models albeit structured data remains the most relevant single data source. As a next step, they test their models with different model heads as depicted in the figure below to explore their respective potentials. Although fully nonlinear models with a DNN as model head generally offer larger modeling flexibility, as they can incorporate interactions, they achieved only slight performance gains over the semi-interpretable models with additive linear model heads. This is particularly interesting as the latter additionally preserve the often desired interpretability of effects. As the semi-interpretable models perform reasonably well, the authors argue that it is indeed possible to obtain interpretable models without losing too much on the performance side.

In the last part of this section, we will allude to several other promising approaches that did pioneering work related to multimodal DL. While most of them use unstructured data sources such as remote sensing data, some do not specifically include structured data. They are still covered in this chapter to give the reader a broad overview of current research in the field. Moreover, structured data could easily be added to each of these models, but often

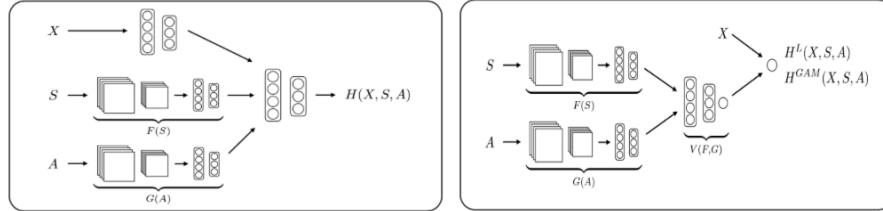


FIGURE 4.6: Fully Nonlinear and Semi-Interpretable Models (X (Struct.) and S+A (Unstruct.) Data (?).

studies intentionally avoid the use of structured data sources as they are sometimes scarcely available due to the cost of data collection. Besides availability, structured data such as household surveys is often irregularly collected and differs vastly between countries, making large scale studies impossible. Therefore, different studies have tried to provide alternatives to classical surveys by applying DL methods on freely available unstructured data sources. While ? use night and daylight satellite images to predict poverty in several African countries, ? use Google Street View images to estimate socioeconomic attributes in the US. Both deploy the classical DL framework such as CNNs to retrieve relevant information from image data for the prediction task. Achieving reasonable prediction results while keeping analysis costs at low levels, both studies outline the potential of their proposed methods as being serious alternatives to current survey based analysis.

Other studies such as ? and ? proposed DL frameworks for satellite imagery in contexts where labelled data is normally scarce. While ? use Deep Gaussian Processes to predict corn yield in the US, ? apply CNNs to detect and map about 516 million buildings across multiple African countries (around 64% of the African continent). Besides being of great importance for applications such as commodity price predictions or financial aid distribution, the results of the two studies could easily be combined with other structured data sources and thereby could constitute a form of multimodal DL with high potential.

4.2.5 Conclusion and Outlook

In the previous sections we have come across various methods of multimodal DL that can deal with both structured and unstructured data. While these often differed substantially in their approach, all of them had in common that they tried to overcome limitations of classical modeling approaches. Examining several of them in detail, we have seen applications of different fusion strategies of data modalities and thereby touched upon related concepts such as end-to-end learning. The issue of interpretability was raised along several

examples by discussing the advantages of different model heads as well as ways to solve identifiability problems using orthogonalization techniques.

It was indeed shown that it is possible to obtain interpretable models that are still capable of achieving high prediction performances. Another finding of past research was that end-to-end learning frequently showed to be superior compared to methods which learn feature representation via independent models or simply retrieve information via expert knowledge. Furthermore, research that actually conducted a comparison between their proposed multimodal DL and single modality models, almost always found their proposed multimodal model to outperform all models which were based on single modalities only. Nevertheless, within the class of single modality models, those using only structured data usually performed best. This leads to the conclusion that structured data often incorporates the most relevant information for most prediction tasks. By contrast, unstructured data sources may be able to add supplementary information and thereby partially improve performances.

While there certainly has been a lot of progress in the field of multimodal DL, conducted analyses still have their limitations which is why results need to be considered with care. Although most research finds their proposed multimodal DL models to achieve excellent performances, not all of them conduct benchmarking with regard to single modality models. Thereby, they limit the possibility to properly evaluate actual improvements over classical modeling approaches. Another aspect that may raise concerns regarding the reliability of results is that multimodal DL models such as most deep learning models have multiple hyperparameters. Together with the flexibility of choosing from a wide variety of data modalities, it opens up the possibility to tune the multimodal models in various ways. Thereby making it possible that actual performance improvements may only be existent for certain configurations of the model as well as combinations of data modalities. This problem is likely to be empathized for studies using only small datasets. Small datasets are especially common in the biomedical context where image data of certain diseases is normally scarce. On top of the previously mentioned aspects, publication bias may be a large problem in the field as multimodal DL models that do not show improvements over single modality or other existing benchmark models, are likely to not be published.

Although there might be concerns regarding the robustness and reliability of some results, past research has surely shown promising achievements that could be extended by future research. While small sample sizes especially for unstructured data such as clinical images were outlined as a great limitation of past research, more of such data will certainly become available in the future. As deep learning methods usually require large amounts of training data to uncover their full potential, the field will probably see further improvements once sufficiently large datasets are available. Hence, including only an increasing number of modalities with limited samples in the models will likely be insufficient. Instead, the most promising approach seems to be incorpo-

rating sufficiently large data amounts of certain unstructured and structured data modalities that contain relevant information for the problem at hand.

4.3 Multi-Purpose Models

Author: Philipp Koch

Supervisor: Rasmus Hvingelby

4.3.1 Intro

After we read about adding further modalities besides video and text, we will look at models that aim to be as generalizable as possible, that not just allow different modalities but also try to achieve multitasking. This section will survey previous attempts for multimodal and multitasking models. Recent developments showed that transformers (Vaswani et al., XXXXb) proved to be the best-suited models for text (BERT, Devlin et al. (XXXXb)) and vision (ViT Dosovitskiy et al. (XXXX)). Thus many recent attempts are also based on the transformer architecture to different degrees. Furthermore, we will also present an early model that is not transformer-based yet found similar behavior in generalizability. We will look further into cutting-edge development in the second half of the section, where the Pathways proposal by Google (Dean, XXXXb) will be revisited, as well as attempts in the broader context of this work. The chapter will conclude with an outlook and a discussion on how the field will likely evolve.

4.3.1.1 Multi-Purpose Models

Since the early years of machine learning, multitask and multimodal learning paradigms have been around. Multitask learning (Crawshaw, XXXX) is the paradigm of training a model on different tasks with the intention that the model transfers the learned knowledge to new tasks, such that fewer resources are required to learn new them. Akin to humans, it is intended that the model benefits from previously learned tasks. Humans do not learn everything new from scratch and spend far fewer resources on learning because knowledge can be transferred. It is thus assumed that deep learning models tend to generalize better when trained on multiple tasks. Multimodal learning (Baltrušaitis et al., XXXX) is a paradigm in which a deep learning model is supplied with multiple modalities like images, text, tables, and so on. As in multitask learning, this approach is also inspired by natural intelligence

since intelligent beings perceive the world through multiple senses. It is thus assumed that multimodal models achieve better performance due to the increased variety of the provided data. We want to marry the two paradigms to form a so-called multi-purpose paradigm for this work. Multi-purpose models are capable of being multimodal and multitask. We assume that this merge improves the generalizability of the models even further.

4.3.2 Previous Work

4.3.2.1 MultiModel

A first prominent multi-purpose model is the so-called MultiModel (Kaiser et al., XXXX). This model, from the pre-transformer time, combines multiple architectural approaches from different fields to tackle both multimodality and multiple tasks. The model consists of four essential modules: the so-called modality nets, the encoder, the I/O Mixer, and the decoder. The modality nets function as translators between the data and a suitable representation for the inner modules. They also follow the purpose of back-translating to create the output. For language tasks, the modality net is a tokenizer that outputs the appropriate embeddings, while for vision tasks, convolution operations transform the images into the proper representation. Furthermore, there are also nets for audio and categorical modalities. The modality net tokenizes the input sequences for the language task, which are then embedded in a unifying vector space and passed on to the encoder. To produce the output, the representations from the decoder are fed into a simple feed-forward network on which a softmax function is used to predict the next token. For the architecture and the chapter generally, it is necessary to recall the concept of Mixture of Experts (MoE) (Jacobs et al. (XXXX), Jordan and Jacobs (XXXX), Shazeer et al. (XXXX)), which is an ensemble of different neural networks inside the layer. The neural networks are not used for every forward pass but only if the data is well suited to be dealt with by a specific expert. MoEs allow for being more computationally efficient while still keeping or even improving performance. Training MoEs usually requires balancing the experts so that routing does not collapse into one or a few experts. An additional gating network decides which of the experts is called. Gating can be implemented so that only K experts are used, which reduces the computational costs for inference. The core model consists of the encoder, the I/O mixer, and the decoder. Input is passed from the modality nets to the encoder first. Subsequently, the encoder passes its output further to the I/O mixer and the decoder. The decoder produces the output sequence. However, producing an autoregressive sequence requires knowledge of the previously generated sequence. Thus the output of the decoder is read by the I/O mixer, which provides the decoder with the necessary information about the previous sequence. The I/O mixer passes its output back to the decoder to create a feedback loop. Both decoder

and I/O mixer require modality nets to read and write in the target domain. The encoder consists of multiple convolution operations and a mixture-of-expert layer block. The I/O mixer and the decoder combine their dual input using cross attention. A positional encoding conceptually similar to the one in transformers was also used for the attention mechanism. MultiModel was trained on eight datasets, from which six were from the language domain and COCO (Lin et al., XXXXc) and ImageNet (Russakovsky et al., XXXX) from the vision domain. For training, four experts in the MoE layers were used. The combined trained MultiModel on ImageNet and machine translation were below state-of-the-art (sota) models. Also, the combined model did not achieve significantly better results than a specialist model, which is the same model but trained solely on one task. However, it was found that the combined model did perform much better on a low-resource task than the respective specialist model. Multimodel offers a pre-transformer approach to deal with different modalities on multiple tasks; although it was only used to generate text, the setup allows extending to other modalities easily.

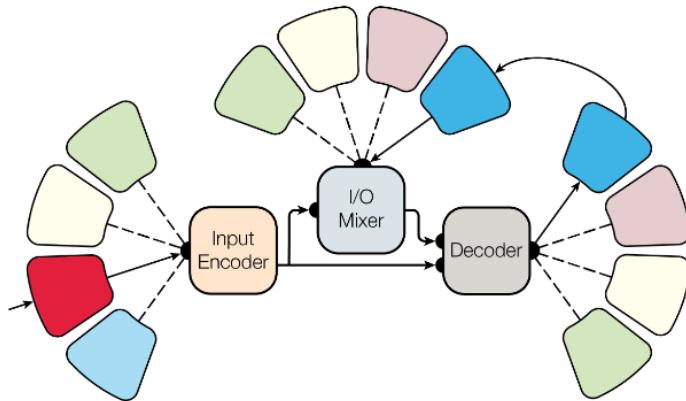


FIGURE 4.7: Architecture of *MultiModel*. The outer boxes without text are the modality nets. From @Kaiser2017.

4.3.2.2 Unified Transformer (UniT)

A more recent multi-purpose model is UniT (Unified Transformer) (Hu and Singh, XXXXb). UniT is built upon the transformer architecture, in which both encoder and decoder are used. To account for multimodality and multitasking, the basic transformer (Vaswani et al., XXXXb) is enhanced. The encoder part of UniT consists of two modality-specific encoders since the initial setup is aimed at the modalities of text and vision. However, more modality-specific encoders may be added. For the case of language, a BERT

model (Devlin et al., XXXXb) is used, while a DETR-encoder model (Carion et al., XXXX) is used for vision. The [CLS] token is also used in the BERT encoder, which is also included in the output sequence. A task-specific token is additionally added to the input of the encoders. The output of the encoders is then concatenated in case all modalities are used. Otherwise, only the modality-specific encodings are passed on. The decoder is modality agnostic and is kept; its input is the concatenated sequence of the encodings and an additional task-specific query. Since the decoder architecture sticks to the DETR model, the decoder does not produce a sequence autoregressively. Instead of taking the previously produced sequence as input, the decoder is fed with the task-specific query vectors instead, thus producing a uniform output. On top of the decoder are task-specific heads needed to transform the decoder output into the desired shape for the specific task. For training, the object detection task requires box loss from DETR, while the other tasks use cross-entropy loss. In experiments, UniT was evaluated against a single-domain version of itself. The general model outperformed the specialist one on multimodal tasks but was outperformed on unimodal tasks by the specialist UniT. UniT was furthermore also outperformed by sota models, although the numbers remained comparable. Even though UniT does not achieve sota or consistently outperforms its specialist version, it is a powerful method to achieve a simple multi-purpose model. By using available encoder models, it is easily extendable and does not require a novel embedding scheme.

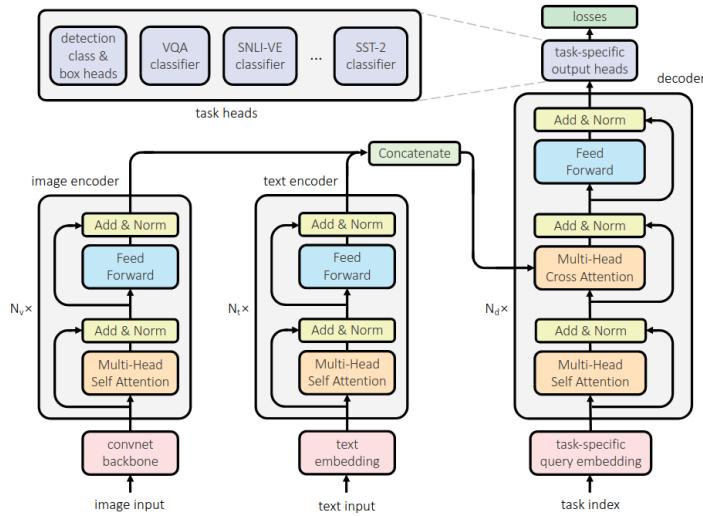


FIGURE 4.8: Modified transformer for UniT. The decoder follows the implementation of DETR [Carion2020]. From Hu2021.

4.3.2.3 OFA - Once For All

Another multi-purpose transformer is OFA (Once For All) (Wang et al., XXXXb). To utilize the sequence-to-sequence (seq2seq) architecture of the transformer, all tasks and the respective modality presented are transformed into seq2seq problems. While other models fuse the modalities in the model, such that the modalities are presented as is to the model, OFA tries a different approach, tokenizing or discretizing all input and embedding it into a shared representation. Since tokenizing an image using a vocabulary is not feasible, a similar approach to ViT (Dosovitskiy et al., XXXX) is used (where input is flattened to 16×16) to obtain a flattened sequence of P representations. These representations are in the same dimension as the token embeddings from text input, which are tokenized using Byte-Pair-Encoding (Sennrich et al., XXXXb). After feeding the embeddings to the encoder, the decoder produces the output as a sequence again. In this case, however, images are represented as a sequence of tokens, similar to the image-patch vocabulary in DALL-E (Ramesh et al., XXXXc). Furthermore, a unique sequence for bounding boxes is also used for object detection and recognition. To generate the task-specific solution, it is thus required that another model is used to generate the images based on the tokens and to visualize the bounding boxes based on the obtained coordinates. Since OFA is an autoregressive model, the objective is based on cross-entropy loss. The probability for the next token is predicted based on the previously produced tokens and the input provided. OFA was trained on different crossmodal tasks: visual grounding, grounded captioning, image-text matching, image captioning, and visual question answering. Further unimodal tasks for training did include: image infilling, object detection, and text reconstruction as in BART (Lewis et al., XXXX). OFA outperformed sota models on cross-modal tasks like image captioning, visual question answering, visual entailment, and visual grounding. On uni-modal tasks, OFA performed well, although it did not outperform sota models. OFA showed additional transfer capabilities to unseen tasks, which were presented with an additional description to solve the task in a few-shot manner. Although the results were satisfactory, the model was not evaluated against a baseline. OFA proved to be a powerful model on the initial transformer architecture. However, a further model might be required to use it specifically for a specific task.

4.3.2.4 Gato - A Generalist Agent

Another model that utilizes the seq2seq approach in transformers is Gato (Reed et al., XXXX). The model can be used as a language model, an agent to play games, and an agent to control robotics. As in OFA, problems are transformed into a seq2seq problem, on which a transformer (decoder only) is applied. Every input from text, vision, robotics, and games is embedded. Visual input is encoded using a flattened sequence of 16×16 patches fed into a

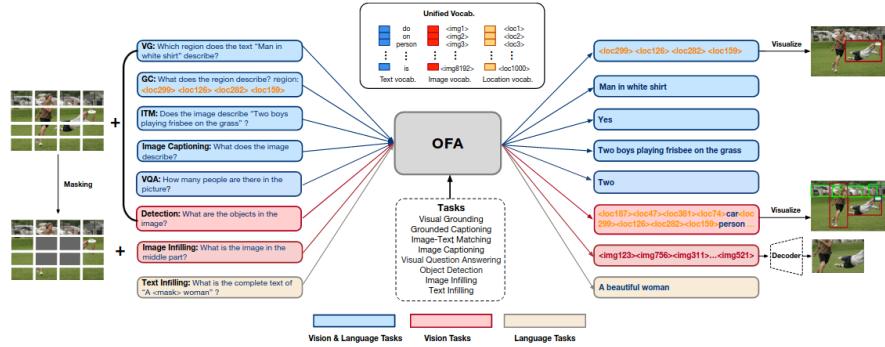


FIGURE 4.9: *OFA*, the different input and output concepts can be seen here. From @Wang2022.

ResNet (He et al., XXXXb), while text input is tokenized using SentencePiece (Kudo and Richardson, XXXX), as done in language models. Furthermore, discrete values like buttons in games and continuous data like movements from robotics are tokenized in a vocabulary of 1024 each too. To represent all modalities sequentially, the different tokens are concatenated. A separator token “|” is added to distinguish the observations from the following action:

$$[x_{\text{Text}}, x_{\text{Images}}, x_{\text{Discrete and Continuous Values}}, |, y_{\text{Action}}]$$

Using this approach, the transformer can predict the next action autoregressively since it is a sequential problem. In the case of text, the action token is also a text token. Since it is only necessary to predict the action based on the previous values, a mask function is added to the cross-entropy loss function, which masks the previous values so that only the next action is predicted and not the input conditioned on. The masking function is always one for text since every previous text token is necessary for language modeling. Gato was evaluated on RL-based tasks against specialist RL agents, where Gato performed worse than the specialist agents or at a human level. On unseen tasks, Gato required fine-tuning since few-shot learning is not feasible due to the input length restrictions in transformers. However, the results were mixed. Some improvements were possible, and the expert was outperformed, but in other cases, massive fine-tuning efforts only led to small gains. It was found that the generalist agent outperformed the specialist agent (only trained for this task) most of the time. Only at Atari Boxing (Bellemare et al., XXXX), Gato was outperformed by the specialist model. Both performed much lower than another task-specific model used as a baseline. In robotics, Gato showed comparable behavior to the baseline sota model. Additionally, Gato also showed capabilities in image captioning and dialogue modeling, although these aspects were not elaborated further.

Similar to OFA, Gato can tokenize all input, whose output can be used to solve the problems Gato is trained on. It was shown that Gato could sometimes transfer knowledge on unseen tasks. However, this did not succeed all over.

4.3.2.5 Comparison

We revisited four multi-purpose models (Multimodel, UniT, OFA, and GATO), of which three were based on the transformer architecture. The last two models transformed the tasks into seq2seq problems to use the standard transformer. UniT is the only model that does not use autoregressive sequence generation, while MultiModel produces the output autoregressively too. Almost all models require some particular module to produce the eventual output: visualizer and image generator in OFA, the task-specific heads in UniT, or the modality nets in Multimodel. On the other side, GATO is more generalizable in its domain since it only produces text and action tokens for games and robotics, which can be translated more easily. None of the models besides OFA achieved sota results. Compared to specialist models, the general models were comparable in their results. Multimodel, OFA, and GATO showed transferability on low-resource or unseen tasks. However, more research in this direction is highly recommended. Multimodel was only compared on a low-resource task against a specialist model, which was trained on, and OFA was not compared to another model for the unseen task. GATO performed better than a specialist model, trained from scratch on most unseen tasks, but failed against the untrained specialist model in Atari Boxing.

4.3.3 Pathway Proposal

Multi-purpose models can also be thought of without the currently well-performing transformer architecture. A proposal for a new approach to multi-purpose models was introduced as Pathways (Dean, XXXXb) aiming to address current pitfalls in deep learning. Current deep learning consists mainly of large dense neural networks trained for a specific task. This paradigm is limiting so that no transfer between tasks is possible, and every time a model is trained, it needs to learn from scratch (or needs to be fine-tuned, where the initial knowledge is lost in the process). Human intelligence can also transfer knowledge between tasks, so it is not necessary to learn everything from scratch. However, transfer in deep learning models is not sufficiently possible, which is a severe constraint.

4.3.3.1 Idea

Pathways follows a different idea to overcome the problems mentioned above. The model consists of a large graph through which data can be forward passed.

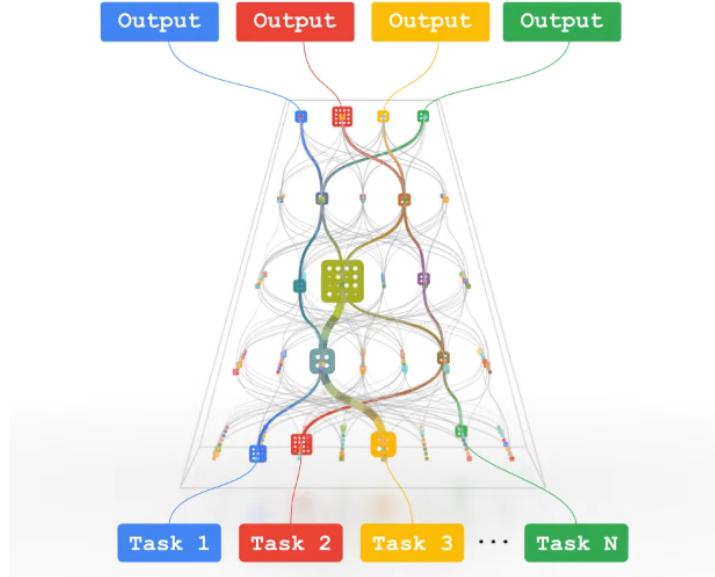


FIGURE 4.10: Concept of pathways. Different tasks follow different paths to different expert models. From @Dean21, [Screenshot August 31th 2022](<https://www.youtube.com/watch?v=Nf-d9CcEZ2w>).

The nodes of the network are neural networks themselves. A pass through this network does not include passing all nodes and thus not all neural networks, but only a few. The pass follows a specific path from one entry to the network's exit. The underlying idea behind this is similar to the mixture-of-expert models described above. Only the specific networks dedicated to solving this problem are to be called. However, pathways consist solely of layers of expert networks. At this point, it is necessary to recall that multitask learning aims to generalize better on new tasks since the knowledge about previously learned tasks can be applied. This idea is the foundation of pathways too, where specialist networks (nodes) are combined in a larger network. It is assumed that the model's generalization capabilities increase significantly by finding an appropriate path for a task to the appropriate expert nodes. In this setup, the particular task-specific problem-solving capabilities are combined. Furthermore, multimodality is also considered a potential extension. Adding more modalities might not be a difficult problem considering the architecture of the previously introduced transformer-based models. Overall the approach of a sparse model combining multiple experts offers many opportunities to combine modalities and reuse task-specific capabilities. The sparsity of the model offers decreased inference time since only some parts of the networks are activated during inference. Another part of the Pathways proposal includes the improvement of current hardware limitations. It is already observable that

Moore’s Law (*each n years, the compute capacity doubles*) has been slowing down substantially, while deep learning research has grown exponentially in the late 2010s (Dean, XXXXa). Thus, hardware also needs to be adapted to the growing demand in deep learning. In the context of the pathway proposal, a novel framework for Google data centers has been introduced, aiming to reduce overhead during computation and access specific parts of the model to utilize the technical advantages of sparse networks. As in dense models, it is not necessary to use the whole network but only chunks of it, which is in contrast to previous dense models, which require the whole model to be accessed. So far, two large pre-trained models have been introduced based on the new training framework. One is the Pathways Language Model (PaLM) [Chowdhery2022], which is currently the largest language model using 540 billion parameters, Minerva (Lewkowycz et al., XXXX). Minerva is based on PaLM, and Parti (Yu et al., XXXX), a text-to-image generator.

4.3.4 Related Work to Pathways

In this section, related architectures will be revisited to provide an overview of how close current research has come to achieving a model like the Pathways proposal.

4.3.4.1 PathNet

An earlier approach for a sparse multitask network, which looks deceptively similar, is PathNet (Fernando et al., XXXX). PathNet is a training algorithm that reuses knowledge from a previously learned task without the risk of catastrophic forgetting (knowledge gets overwritten), thus using the positive aspects of multitask learning. The objective of PathNet is built upon an evolutionary algorithm (EA). An evolutionary algorithm is used to optimize a discrete problem where derivative-based algorithms cannot be applied. The algorithm is based on a population (in the domain to be optimized) and a fitness function (can be used to evaluate how close a member of the population is to the optimum). Parts of the population are chosen to create offspring either by mutation or recombination. The resulting population is then evaluated with respect to their fitness function, and only the best-suited individuals are kept. The same procedure is repeated based on the resulting population until a specific criterion is met (e.g., convergence). While evolving the population, it is necessary to balance exploration and exploitation to find the desired outcome. Since EAs are research topics themselves and may vary heavily, we refer to (Bäck and Schwefel, XXXX) and, more recently, to (Doerr and Neumann, XXXX) for further insights. Neural networks are often depicted as a graph in which the input is directed to all nodes in hidden layers, and their output is again passed to all nodes in the next hidden layer or an output layer. In the case of PathNet, each node is itself a neural network; thus, all data is

passed to each node. The training algorithm finds the best paths for a specific task through the network. At first random paths through the network are initialized, then the paths are trained for T epochs. After training, the paths are evaluated against each other, where the winning path overwrites the losing path. To achieve exploration, however, the overwritten path is mutated. The path is intended to deviate a little by also visiting neighboring nodes to achieve exploration. Until a specific criterion (e.g., number of epochs) to stop is reached, the current paths are frozen so that no more modifications to the parameters of the networks on this path are possible. All other parameters are newly initialized again. Also, a different, task-specific head is initialized. The same procedure is now done again for another task. The main difference is that the previously obtained path, including the trained networks, is frozen during training so that the model can transfer knowledge from the previous task to the new task. The model finds appropriate paths throughout the network until the stopping criterion is met. PathNet was evaluated on supervised learning tasks and on reinforcement learning scenarios. Learning from scratch and fine-tuning a PathNet, trained on the first task, were chosen as a baseline. Fine-tuning is ran on top of the first model, which PathNet trains for the first task. Overall, PathNet improved training time and prediction quality for the second task compared to standard fine-tuning and learning from scratch. PathNet has shown that positive transfer in a Pathways-alike architecture is possible. Different tasks can reuse the knowledge from training on previous tasks.

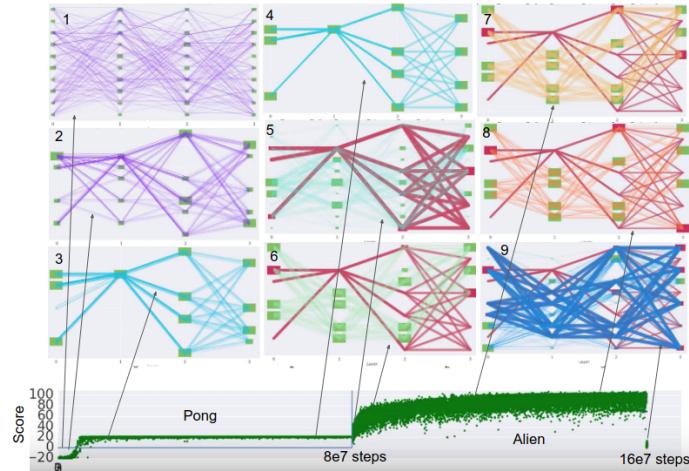


FIGURE 4.11: Training *PathNet* on two tasks. At first random paths are initialized (1), then trained (2-3) and fixed (4). The same procedure is repeated for the next paths using the previously fixed paths and new parameters in all other nodes (5-9). From @Fernando2017.

4.3.4.2 LIMoE

In the context of Pathways, the work LIMoE (Multimodal Contrastive Learning with LIMoE: the Language-Image Mixture of Experts) (Mustafa et al., XXXX) is also necessary to mention since it achieves a multimodal sparse model. LIMoE combines text and vision input using a transformer encoder, which includes an MoE layer that can be used sparsely during inference time. Besides the context of the Pathways proposal, LIMoE also aims to provide a new approach to contrastive learning. While previous methods used two models (two-tower) to encode the modalities, LIMoE is solely based on one model, where the modalities are processed in a single transformer-like model (one-tower). Thus, the model described in LIMoE is based on a modified transformer. The text data is encoded using one-hot-SentencePiece(Kudo and Richardson, XXXX) encoding, while images are tokenized in the same way as in ViT (Dosovitskiy et al., XXXX) to provide the input appropriately. The main difference to the standard transformer is an MoE layer where the feed-forward network usually lies. In this layer, E experts are used, which are themselves feed-forward-networks. For each token, K appropriate experts will map the tokens further downstream. The routing is computed by a gating net network, which decides which K experts are called. Another feature here is a fixed-length buffer for each expert in the MoE layer. This buffer is used to store tokens before an expert network processes them, assuming that the allocations of tokens for each expert are balanced. If it is impossible to buffer tokens for the experts, the tokens will be dropped. To process the more important tokens first, Batch Priority Routing (Riquelme et al., XXXX) is used to provide a ranking mechanism. The output of the transformer encoder is then average pooled and subsequently multiplied with a modality-specific weight matrix, which produces the eventual output for the token of both modalities.

The model is trained using a contrastive objective. The contrastive loss, in this case, aims to maximize the paired visual and textual input while minimizing all combinations of unpaired embeddings. This objective can be achieved by using the dot-product as a similarity measure between the two embeddings of both modalities, which provide a differentiable operation through which the overall loss can be minimized. Additionally, to contrastive loss, the pitfalls of a multimodal MoE are also considered. One challenge in MoE is the correct balancing of routing to the experts, which is even more challenging when using unbalanced multimodal data. To address this issue, two new losses based on entropy are introduced. Entropy can be used as an appropriate term since it provides a valuable number for the uniformity of the distribution, which is necessary to balance the expert assignments. The losses are aimed at controlling the allocation of experts to tokens, which is also necessary to fulfill the assumptions for the implemented buffer. One loss considers the token level (local) routing distribution, and the other considers the overall expert routing distribution (global). The local loss aims to achieve no uniform behavior in expert allocation such that each token is indeed assigned to specific ex-

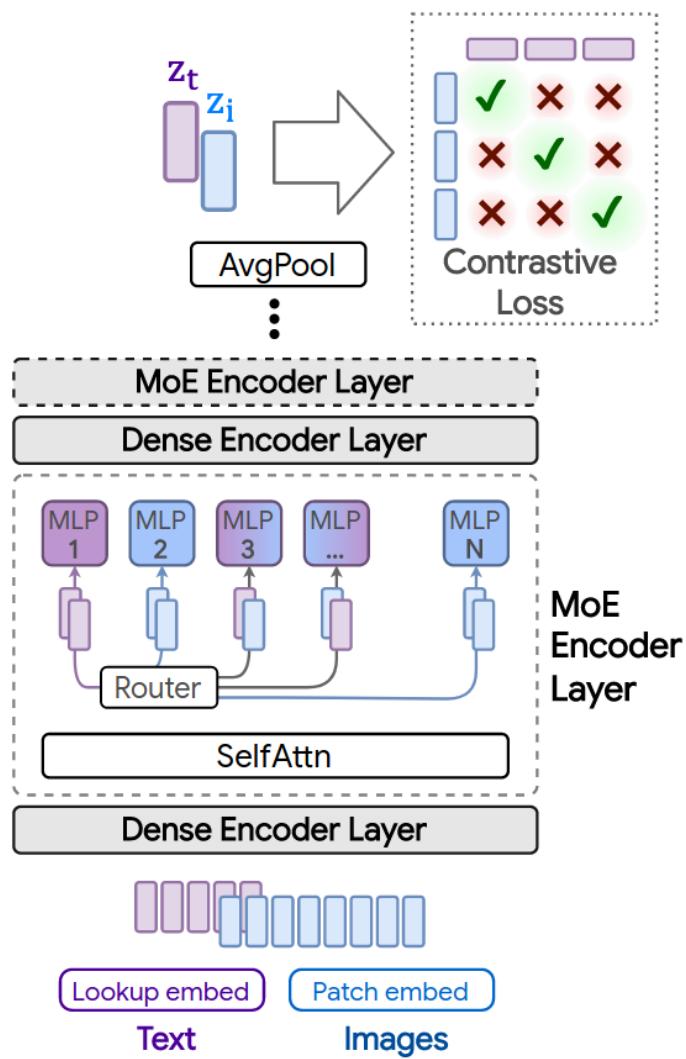


FIGURE 4.12: Architecture of *LIMoE*. From @Mustafa2022.

perts. In contrast, the overall global loss aims to achieve uniformity over all tokens to avoid a collapse in which tokens are solely assigned to a few experts which do not have the capacity to deal with all tokens. These losses are computed for each modality. Furthermore, already available losses for training MoE models were also added to avoid known downsides of MoE models. LIMoE was compared against similar models, which show the pitfalls of being solely dense and specialized on a task, which is also the case for two-tower setups like CLIP (Radford et al., XXXXa). The test dataset was ImageNet (Deng et al., XXXX) and COCO (Lin et al., XXXXc). Overall, LIMoE-H/14 (largest model, 12 MoE-layers, 32 experts per layer) achieved strong performance considering that only one model was used for two modalities against specialist models in two-tower setups. It was also possible to outperform CLIP by a significant margin while using minimal additional parameters. Models that achieved similar results to LIMoE used at least twice the number of parameters for a forward pass. So far, LIMoE is a new approach to contrastive learning. It showed that the sparse approach could significantly lead to strong performance while using fewer parameters for inference. Furthermore, it also includes essential aspects of the Pathways proposal using the MoE layer where sparsity comes into play. The authors stated that the critical aspect is how the model routes to the specific expert, which might also be helpful in an additional multitask setup, which is an important step towards a Pathways-alike model. However, the authors also stated that merging modalities remains a field of research, as well as adding further modalities. Since Vision and text have already proven to be non-trivial to combine, more problems might arise with further modalities.

4.3.4.3 muNet (Multitask Network)

Another model which is worth considering in the context of the Pathways proposal is muNet (Gesmundo and Dean, XXXX). The initial criticism of current inefficiencies in deep learning is addressed by providing a framework to avoid discarding knowledge. muNet is an approach to improving multitask learning with a focus on fine-tuning. In current practice for fine-tuning, a pre-trained model is copied and then explicitly trained on a task by overwriting previous knowledge, which addresses the same problem outlined by Dean (XXXXb). muNet evolves a pre-trained model on different tasks, using an EA to reuse already trained knowledge. An initial model is modified so that it becomes more and more optimized to fit specific tasks while keeping all previously learned knowledge. These modifications are so-called mutations applied to the model to score better. Eventually, a set of models is obtained, which includes new neural networks based majorly on the parameters of the initial model. The new modules can be seen as paths to task-specific modifications of the initial network. Although the network is not sparse, the concept of reusing many parts of a network while accounting for multiple tasks is to be mentioned in the context of the Pathways proposal. The EA of muNet starts with an ini-

tially proposed model that is mutated further on. All further mutations are stored so that after a set of candidates is available, the set can be split into models trained for this task (active population) and models for other tasks (inactive population). These two sets become the sets of candidates for the following task-specific iterations. Training a specific task follows three steps: sampling candidate models, mutating, training, and evaluation. The best scoring model is added to the active population for further mutation. A sample algorithm accounts for exploration and exploitation to get a candidate model for subsequent mutation. The active population is ordered in a descending list based on the model's score. Each list entry is then revisited, starting from the highest scoring model onward, so that the better performing models are considered first (exploitation). The draw probability is computed as:

$$P(m|t) = 0.5^{\#timesSelected(m,t)}$$

where:

$$\#timesSelected(m,t)$$

is the amount of previous mutations based on model m for task t). The more unsuccessful mutations the model has had before, the smaller the draw probability becomes. Thus, exploration is emphasized by considering previous attempts and allowing other models to be preferred. However, if this method does not yield a candidate, a model is drawn from the union of the inactive and active population. Applying mutations is the next step in the algorithm. A random number of mutations are drawn from the set of possible mutations, which include:

- Layer cloning, a layer is cloned for training. The layer's parameters are copied from the parent model so that training can continue using the same knowledge. The other layers are still used but are not updated. Additionally, the task-specific head layer is cloned to account for the underlying changes. In the case of training on a new task, the head is also newly initialized.
- Layer insertion. Two layers are added to the model as residual adapters ((Rebuffi et al., XXXX), [Houlsby2019@]). The second layer is zero-initialized to keep an identity function so that training can continue from the state before mutation.
- Layer removal is used to skip layers while still using all other layers of the parent model in a frozen state.
- Hyperparameter change: samples hyperparameters close to the ones of the parent model. A list of neighboring values is constructed from which a parameter is drawn.

Subsequently, the models are trained on the task and scored. If the mutated model is better than the parent model, it is also added to this task's set of active models. This routine is done for all tasks iteratively and can be repeated several times. Ultimately, only the best scoring models are kept for each task, yielding a list of models consisting of paths to layers of the old

model and newly trained ones. muNet was evaluated for fine-tuning against a ViT instance, which is aimed at being the most generalizable (Steiner et al., XXXX). The evaluation benchmarks consisted of multiple classification problems (to simulate multitasking). ViT was fine-tuned on all of these tasks as a baseline. In contrast, another ViT was evolved using muNet, on which the baseline model was evaluated. The approach using muNet outperformed the fine-tuned ViT while using significantly fewer parameters. Although muNet is neither a multi-purpose model nor a sparse model, the concept of building task-specific paths by enhancing pre-trained models fits in the broader context of the Pathways proposal. Research in the direction of combining automated routing for tasks and adding multimodality might lead toward the vision of Pathways and is thus highly recommended.

4.3.5 Conclusion Pathways

So far, steps have been made to fulfill the proposed direction of the Pathways paper. Hardware upgrades were already introduced, on which sota models were published (PaLM, Parti, and Minerva). Moreover, steps towards sparsity (LIMoE) and research into minimal resource-intensive deep learning, as outlined in muNet, were made. However, many questions remain open. How will the particular architecture of a Pathways-like model be structured? Will it have a layer-like structure as PathNet and as shown in the proposal (Dean, XXXXb)? Will the paths follow a hierarchical structure? What will the sparsity look like in the model itself? How exactly will routing be achieved? Will EA play a more significant role?

The previously mentioned successes in multitask learning, hardware, and the combination of MoE and multimodal, as seen in LIMoE, already hint that this direction is likely to be researched further in the upcoming month.

4.3.6 Discussion

Most of the developments described in this section took place in the last two years. The Pathways proposal is not a year old, and research in this direction is ongoing, with more and more results to show. Thus, novel models will likely be introduced in the upcoming months, which will probably set new sota results. NLP shows that the larger models get, the better they perform (Chowdhery et al. (XXXXb), Brown et al. (XXXXa)), which might also lead to the same trend for multi-purpose models.

A pitfall of models of these sizes is the low accessibility. Researchers need to access the model through an API since running these models on a few GPUs will be highly unlikely. It might be unlikely to see a BERT-like engagement with the community of researchers if the access to models remains limited.

On the contrary, more open-source collaborations, as seen with [EleutherAI](#) or [Huggingface](#), might evolve as well as a countermovement and techniques like distillation ([Hinton et al., XXXX](#)) might become more critical. Another issue with multi-purpose models is the lack of metrics. Current metrics are not suited for multitask and multimodal models. Evaluation might also become harder since many different modalities can be used, as seen here with the robotics property of GATO, which was not used in any of the models. Eventually, it is also necessary to consider the societal impact. The bias problem will also become an issue in multi-purpose models, especially since multiple datasets must be considered now. Also, the environmental impact of training large models needs to be considered since it is likely that larger models will yield better performance according to scaling laws ([Reed et al., XXXX](#)) but will also have a larger carbon footprint.

4.4 Generative Art

Author: Nadja Sauter

Supervisor: Jann Goschenhofer



FIGURE 4.13: LMU logo in style of Van Gogh's Sunflower painting

As we have seen in subsection 3.2, with the help of multimodal deep learning computers can create images only based on text prompts. This new ability is also used in arts in the field of ‘generative art’ or also known as ‘computer art’. The new movement comprises all artwork where the human artist cedes control to an autonomous system ([Galanter, XXXX](#)). In this way everyone, even artistically untalented people, can easily create pictures as the computer does the work for you. In some way, the computer becomes the artist with some sort of creativity, a distinct human ability. In this chapter we want to give an overview about how computers improved over time in generating images

and how this is used in the art scene. For instance in Figure 4.13 we used the seal of the Ludwig Maximilians Universitiy and changed the style to Van Gogh's [Sunflower painting](#) by the [Neural Style Transfer Algorithm](#) and the method [CLIP + VQGAN](#).

4.4.1 Historical Overview

The first attempt to use AI to generate pictures was made by Google engineer Alexander [Mordvintsev \(XXXX\)](#) and his "DeepDream" Software. He used Convolution Neural Networks to generate very interesting and abstract images based on the activation of a layer, visualizing the patterns learned by a neural network. Below you can see a picture of a Labrador after processed by the DeepDream algorithm.



FIGURE 4.14: Picture of a Labrador processed by DeepDream ([Google Colab](#))

In the following year, [Gatys et al. \(XXXX\)](#) investigated methods to transfer the style of pictures. This method was used to transfer the style of Van Gogh's Sunflower painting to the LMU seal in the beginning of this chapter (see Figure 4.13). Besides, below in Figure 4.15 you can see the same Labrador picture from Figure 4.14 in [Kandinsky style](#).

Furthermore, the architecture of Generative Adversarial Networks (GANs), which was first introduced by Google Researcher [Goodfellow et al. \(XXXXa\)](#), was used by NVIDIA researches [Karras et al. \(XXXX\)](#) to create very realistic fake images with their architecture StyleGAN. For instance, you can create pictures of people who do not exist, but look totally realistic (see Figure 4.16).

Nevertheless, it was almost not possible to control the exact output of these early forms of AI art. There was no option to make specifications of how the result should look like in detail. For instance, you always get a human face with the earlier mentioned StyleGAN application, but you cannot specify to



FIGURE 4.15: Picture of a Labrador with Kandinsky style ([Google Colab](#))

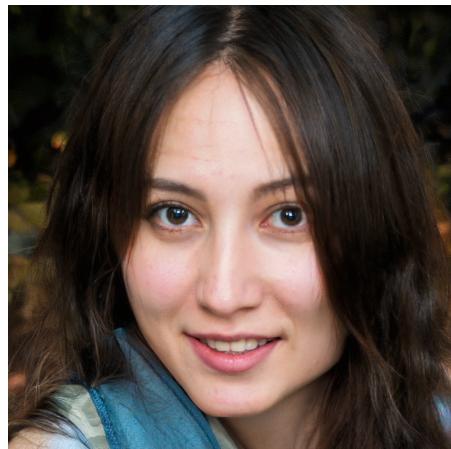


FIGURE 4.16: Fake face generated by [StyleGAN](#)

generate a blond girl with blue eyes. This can be achieved by applying the artist-critic paradigm ([Soderlund and Blair, XXXX](#)): First of all, the computer as artist generates a picture based on what the Neural Network learned in the training phase (e.g. StyleGAN learns to generate pictures of human faces). Additionally, you also need a critic that tells the computer if the output satisfies the concrete idea of the human artist. For this reason multimodal deep learning models emerged in the field of generative art. Here you can control the output with the help of text prompting. In this way you can check if the generated picture matches the initial text description. Looking at our StyleGAN example, the multimodal architecture supervises if the output picture is indeed a blond girl with blue eyes. A new class of models for generating pictures evolved.

This idea was used by OpenAI for their models DALL-E ([Ramesh et al.,](#)

XXXXb) and CLIP (Radford et al., XXXXb) which were released in January 2021. Both architectures are critics for multimodal models. Only a few days after the release, Ryan Murdock combined CLIP (critic) with the already existing Neural Net “BigGAN” (artist) in his “The Big Sleep” software. Furthermore, Patashnik et al. (XXXX) developed StyleCLIP, a combination of StyleGAN (artist) and CLIP (critic) to edit parts of images via text instructions. In the following months, Katherine Corwson combined CLIP with the existing VQGAN algorithm. She also hooked up CLIP with guided diffusion models as artists. This approach was further investigated by OpenAI that published a paper (Dhariwal and Nichol, XXXX) in May 2021 about guided diffusion models. Moreover, in December 2021 they introduced GLIDE (Nichol et al., XXXX), a model with CLIP or classifier-free guidance as critics and diffusion models as artists. For more technical details about text2img methods like DALL-E and GLIDE look at subsection 3.2 or for text supporting CV models like CLIP at subsection 3.4.

4.4.2 How to use models?

A lot of different notebooks are publicly available to apply the different pre-trained models. In general, all notebooks work pretty similar: You only need to enter your text prompt in the code and after running the notebook the computer generates a picture based on your instruction. It is easy and no coding knowledge required. Moreover, there are also some API and GUI applications (e.g. [MindsEye beta](#)), and in this way no programming is needed at all. Using these models it is important to think about how exactly you enter your text prompt. You can influence the output in a desired way with little changes in your short text instruction. This is also known as prompt engineering. For instance, in the beginning of this chapter, we entered “in the style of Van Gogh” to change the style of the LMU seal. Besides, a special trick is to append “unreal engine” ([Aran, XXXX](#)) which makes picture more realistic and higher in quality. This seems surprising, but the models were trained on data from the internet including pictures of the software company Epic Games that has a popular 3D video game engine called “Unreal Engine”. This is one of the most popular prompting tricks.

Unfortunately, OpenAI has never released DALL-E. There is only an open-source version called ruDALL-E ([Shonenkov, XXXX](#)) that was trained on Russian language data. Besides, hugging face hosts DALL-E mini ([Boris, XXXX](#)) where you can generate pictures, but do not have access to the model itself. PyTorch offers a replication of the DALL-E code ([OpenAI, XXXX](#)) but no trained model. Furthermore, CLIP was released without publishing the used training data. However, there exists an open source data set with CLIP embeddings called LAION-400m ([Schuhmann et al., XXXX](#)). In the following, we used different publicly available notebooks to try out the different models [CLIP + BigGAN](#), [CLIP + VQGAN](#), [CLIP + Guided Diffusion](#), [GLIDE](#) with

the text prompt “*a fall landscape with a small cottage next to a lake*” (see Figure 4.17) and “*panda mad scientist mixing sparkling chemicals, artstation*” (see Figure 4.18). The first prompt shows pretty realistic results, whereas the second prompt results in more different and crazier outputs. That is because the panda-prompt is more abstract than the first one and in this way more difficult to illustrate. In addition, some of the notebooks run on lower resolution due to computational limitations. Besides, GLIDE is also downsized by the publisher: The released smaller model consists of 300 million parameters, whereas the unreleased model has about 3.5 billion parameters (Nichol et al., XXXX). So better results are possible with higher computational power and other implementations of the models.

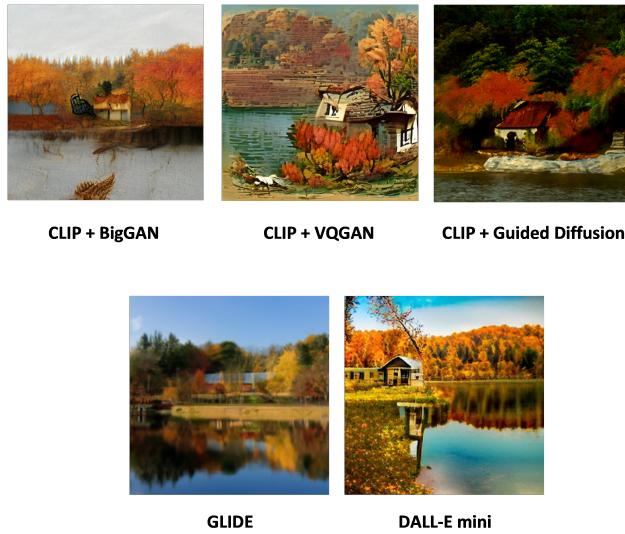


FIGURE 4.17: Comparison of different models with prompt “fall landscape with a small cottage next to a lake”

4.4.3 Different tasks and modalities

So far, we concentrated on the two modalities text and image. Combining both of them, you can tackle different tasks with the models mentioned above. The main usage is to generate images based on a text prompt. Therefore, you can start from noise or you can choose a real image as starting point (Qiao et al., XXXX). This was done in the beginning with the LMU seal by CLIP + VQGAN (see Figure 4.13): instead of starting from noise, the model started from the LMU seal as initialization and then used the prompt “in style of Van Gogh”. The video captures how the model develops during fitting. In the end



FIGURE 4.18: Comparison of different models with prompt “panda mad scientist mixing sparkling chemicals, artstation”

the typical Van Gogh sunflowers emerge as well as what could be a part of Van Gogh’s face.

Besides, you can edit, extend, crop and search images with models like GLIDE (Nichol et al., XXXX). For instance, Nichol et al. (XXXX) fine-tuned the model for text-conditional image inpainting (see figure 4.19). By marking some area in the pictures, here in green, and adding a text prompt, you can edit pictures very easily and precise. This is quite impressive as the model needs to understand from the text prompt which object should be filled in and then do this in the right style of the surrounding to produce a realistic outcome. Another idea is to use a sketch of a drawing and let the model fill in the details based on a text caption (see figure 4.20 below). This allows controlled changes of parts of pictures with relatively little effort. In this way GLIDE can be used to generate pictures out of nothing, but also to edit pictures in a specific way. Furthermore, it is also possible to combine other modalities as well (see more details in subsection 4.1). For instance, WZRD (XXXX) accompanies your video with suitable audio. Or it is even imaginable to create sculptures with 3D-printers (McCormack and Gambardella, XXXX).



FIGURE 4.19: Text-conditional image inpainting examples with GLIDE (Nichol et al., XXXX)



FIGURE 4.20: Text-conditional edit from user scratch with GLIDE (Nichol et al., XXXX)

4.4.4 Discussion and prospects

In the last years methods to generate images via text prompting improved tremendously and a new field in arts arised. It is surprising how these models are able to create images only based on a short text instruction. This is quite impressive as AI achieved some kind of creativity. It is up for discussion to which extent the computer is becoming the artist in generative arts and in this way replacing the human artist. However, there is still no direct loss function that can calculate how aesthetically pleasing a picture is (Esser et al., XXXXa). This is probably also quite subjective and cannot be answered for everyone in the same way. Most of the time the computer works as aid for the creative process by generating multiple images. Then, the human artist can pick the best outcome or vary the text prompt to improve the output in a desired way. However, the better AI becomes, the less the human artist needs to intervene in this process.

Furthermore, as the output becomes more and more realistic, there is the risk that these methods are abused to facilitate plagiarism or create fake content and spread misleading information (Dehouche, XXXX). After all, the

output looks totally realistic, but are completely made-up and generated by the computer. For this reason, some organisations like Open-AI do not release all their models (e.g. DALL-E) or downstream models (e.g. CLIP). On the other hand from a scientific point of view, it is important to get access to such models to continue research.

Moreover, similarly to most Deep Learning algorithms, these models are affected by biases in the input data ([Srinivasan and Uchino, XXXX](#)). For instance, [Esser et al. \(XXXXa\)](#) points out that CLIP text embeddings associate a human being more with a man than a woman. In this way it might be likelier that our models generate a man with the text prompt “human being” than a woman. This effect needs to be further investigated and should be removed.

After all, generative arts can be used to create Non Fungible Tokens (NFT) really easily. NFTs are digital artworks where a special digital signature is added making them unique and in this way non-fungible ([Wang et al., XXXXc](#)). The digital artwork is bought and sold online, often by cryptocurrency. That is why this field is also called Cryptoart. This provides the perfect platform to sell generative arts. However, this trading market is quite new and controversial like cryptocurrency in general.

In conclusion, generative arts is quite a new and impressive field. It combines technology with arts, two rather opposite fields. The methods are already really impressive and still getting better and better. For instance, this year Open AI already published DALLE-2 ([Ramesh et al., XXXXa](#)) that outperforms DALLE-1. It remains highly interesting to follow up the developments in this field.

5

Conclusion

Author: Nadja Sauter

Supervisor: Matthias Assenmacher

It is very impressive how multimodal architectures developed especially over the course of the last two years. Particularly, methods to generate pictures based on text prompts like DALL-E became incredibly good in their job. A lot of people are fascinated by the stunning results and a huge hype about these AI generated images evolved in the internet, especially on twitter. In this way, the models were not only investigated by researchers but also by the AI online Community (e.g. Katherine Crowson alias [Rivers Have Wings](#)). Even in the art scene these methods attracted a lot of attention as shown in our use case “Generative Arts” (subsection 4.4). Apart from that, it is possible to deploy these methods commercially, for instance in the film production or gaming industry (e.g. creating characters for games). However, this also results into problems of copyright.

It is impressive how realistic and precise outputs are achieved by such architectures. On the other hand, these methods can also be abused to spread misleading information as it is often very difficult to distinguish between a fake or a real picture by only looking at it. This can be systematically used to manipulate the public opinion by spreading AI manipulated media, also called deep fakes. That’s why researchers like Joshi et al. ([XXXX](#)) demand automated tools which are capable of detecting these fabrications. Apart from that, like most Deep Learning models, multimodal architectures are not free from bias which also needs to be investigated further ([Esser et al., XXXXa](#)). Besides, the algorithms are very complex which is why they are called “black-box” models, meaning that one cannot directly retrace how the model came to a certain solution or decision. This may limit their social acceptance and usability as the underlying process is not credible and transparent enough ([Joshi et al., XXXX](#)). For instance, in medical applications like e.g. predicting the presence or absence of cancer, apart from the decision of the AI the reasoning and the certainty are highly relevant for doctors and patients.

Furthermore, there is a clear trend in recent years to build more and more complex architectures in order to achieve higher performance. For instance OpenAI’s language model GPT-2 has about 1.5 Billion parameter ([Radford et al., XXXXc](#)) whereas its successor GPT-3 has about 175 Billion parameters

(Brown et al., XXXXb). Increasing the number of parameters often helps to improve model performance, but all of these parameters need to be trained and stored which takes a lot of time, enormous computational power and storage. For example, training GPT-2 took about one week (168 hours) of training on 32 TPUv3 chips (Strubell et al., XXXXb). The researchers Strubell et al. (XXXXb) estimated that the cloud compute costs for training GPT-2 add up to about \$12,902–\$43,008. Apart from the enormous expenses, this also contributes to our environmental burden as this process is really energy intensive. Due to missing power draw data on GPT-2’s training hardware, the researchers weren’t able to calculate the CO₂ emission. However, for the popular BERT architecture with 110M parameters they calculated cloud compute costs of \$3,751–\$12,571, energy consumption of 1,507 kWh and a Carbon footprint of 1,438 lbs of CO₂. In comparison, the footprint of flying from New York to San Francisco by plane for one passenger is about 1,984 lbs of CO₂. In conclusion training BERT once results in almost the same footprint as this long-haul flight. On top of this, these numbers are only for one training run. Developing a new model or adapting it often takes several fitting and tuning phases.

Moreover, the computational power as well as the necessary hardware, technology and financial means to run these models can only be provided by big technology companies like e.g. Google, Facebook or Open AI. This results in a disparate access between researchers in academia versus industry. Furthermore, the companies often do not publish their models as they are their ‘product’ and intellectual property. In this way it is not possible to reproduce their work and findings independently. Besides, from an economic point of view, this may be the foundation of a monopoly which can be dangerous for economic competition and could be abused by the Tech Giants.

6

Epilogue

Author:

Supervisor:



7

Acknowledgements

The most important contributions are from the students themselves. The success of such projects highly depends on the students. And this book is a success, so thanks a lot to all the authors! The other important role is the supervisor. Thanks to all the supervisors who participated! Special thanks to [Christian Heumann](#) and [Bernd Bischl](#) who enabled us to conduct the seminar in such an experimental way, supported us and gave valuable feedback for the seminar structure. Thanks a lot as well to the entire [Department of Statistics](#) and the [LMU Munich](#) for the infrastructure.

The authors of this work take full responsibilities for its content.



Bibliography

- Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., and Soroa, A. (XXXX). A study on similarity and relatedness using distributional and wordnet-based approaches.
- Ailem, M., Zhang, B., Bellet, A., Denis, P., and Sha, F. (XXXX). A probabilistic model for joint learning of word embeddings from texts and images.
- Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (XXXXa). Flamingo: a visual language model for few-shot learning.
- Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. (XXXXb). Flamingo: a visual language model for few-shot learning.
- Alford, A. (XXXX). Google announces 800m parameter vision-language ai model align.
- Anderson, P., Fernando, B., Johnson, M., and Gould, S. (XXXXa). Spice: Semantic propositional image caption evaluation. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 382–398. Springer International Publishing.
- Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (XXXXb). Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.
- Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. (XXXX). Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.
- Aran, K. (XXXX). When you generate images with vqgan clip, the image quality dramatically improves if you add "unreal engine" to your prompt. people are now calling this "unreal engine trick".

- Bäck, T. and Schwefel, H.-P. (XXXX). An overview of evolutionary algorithms for parameter optimization. 1(1):1–23.
- Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. (XXXXa). Data2vec: A general framework for self-supervised learning in speech, vision and language.
- Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (XXXXb). wav2vec 2.0: A framework for self-supervised learning of speech representations. 33:12449–12460.
- Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (XXXX). Multimodal machine learning: A survey and taxonomy. 41(2):423–443.
- Bandy, J. and Vincent, N. (XXXX). Addressing” documentation debt” in machine learning research: A retrospective datasheet for bookcorpus.
- Banerjee, S. and Lavie, A. (XXXX). 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. Association for Computational Linguistics.
- Bao, H., Dong, L., and Wei, F. (XXXX). Beit: Bert pre-training of image transformers.
- Barham, P., Chowdhery, A., Dean, J., Ghemawat, S., Hand, S., Hurt, D., Isard, M., Lim, H., Pang, R., Roy, S., Saeta, B., Schuh, P., Sepassi, R., Shafey, L. E., Thekkath, C. A., and Wu, Y. (XXXX). Pathways: Asynchronous distributed dataflow for ml.
- Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (XXXX). The arcade learning environment: An evaluation platform for general agents. 47(1):253–279.
- Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. (XXXX). Are we done with imagenet?
- Birhane, A., Prabhu, V. U., and Kahembwe, E. (XXXX). Multimodal datasets: misogyny, pornography, and malignant stereotypes.
- Boris, D. (XXXX). Dall·e mini.
- Bosch, A., Zisserman, A., and Munoz, X. (XXXX). Image classification using random forests and ferns. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. Ieee.
- Bowman, S. R. and Dahl, G. E. (XXXX). What will it take to fix benchmarking in natural language understanding?

- Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (XXXX). Signature verification using a "siamese" time delay neural network. 6.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (XXXXa). Language models are few-shot learners. 33:1877–1901.
- 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. (XXXXb). Language models are few-shot learners.
- Bruni, E., Tran, N.-K., and Baroni, M. (XXXX). Multimodal distributional semantics. 49:1–47.
- Brysbaert, M., Warriner, A. B., and Kuperman, V. (XXXX). Concreteness ratings for 40 thousand generally known english word lemmas. 46(3):904–911.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (XXXX). End-to-end object detection with transformers.
- Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (XXXX). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660.
- Cheerla, A. and Gevaert, O. (2019). Deep learning with multimodal representation for pancancer prognosis prediction. *Bioinformatics*, 35(14):i446–i454.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (XXXXa). A simple framework for contrastive learning of visual representations.
- Chen, X., Xie, S., and He, K. (XXXXb). An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649.
- Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (XXXXa). Palm: Scaling language modeling with pathways.

- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsyyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. (XXXXb). Palm: Scaling language modeling with pathways.
- Collell, G., Zhang, T., and Moens, M.-F. (XXXX). Imagined visual representations as multimodal embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Cornia, M., Stefanini, M., Baraldi, L., and Cucchiara, R. (XXXX). Meshed-memory transformer for image captioning.
- Crawshaw, M. (XXXX). Multi-task learning with deep neural networks: A survey.
- Das, A., Agrawal, H., Zitnick, L., Parikh, D., and Batra, D. (XXXX). Human attention in visual question answering: Do humans and deep networks look at the same regions? 163:90–100.
- Dean, J. (XXXXa). 1.1 the deep learning revolution and its implications for computer architecture and chip design. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 8–14.
- Dean, J. (XXXXb). Introducing pathways: A next-generation ai architecture.
- Dehouche, N. (XXXX). Plagiarism in the age of massive generative pre-trained transformers (gpt-3). 21:17–23.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (XXXX). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Devereux, B. J., Tyler, L. K., Geertzen, J., and Randall, B. (XXXX). The centre for speech, language and the brain (cslb) concept property norms. 46(4):1119–1127.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (XXXXa). Bert: Pre-training of deep bidirectional transformers for language understanding.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (XXXXb). Bert: Pre-training of deep bidirectional transformers for language understanding.

- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (XXXXc). BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186. Association for Computational Linguistics.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (XXXXd). Bert: Pre-training of deep bidirectional transformers for language understanding.
- Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K.-W., and Gupta, R. (XXXX). Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 862–872.
- Dhariwal, P. and Nichol, A. (XXXX). Diffusion models beat gans on image synthesis.
- Doerr, B. and Neumann, F. (XXXX). A survey on recent progress in the theory of evolutionary algorithms for discrete optimization. 1(4).
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (XXXX). An image is worth 16x16 words: Transformers for image recognition at scale.
- Esser, P., Rombach, R., and Ommer, B. (XXXXa). A note on data biases in generative models.
- Esser, P., Rombach, R., and Ommer, B. (XXXXb). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.
- Everingham, M., van Gool, L., Williams, C., Winn, J., and Zisserman, A. (XXXX). The pascal visual object classes (voc) challenge. 88(2):303–338.
- Fellbaum, C. D. (XXXX). Wordnet : an electronic lexical database. 76:706.
- Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel, A., and Wierstra, D. (XXXX). Pathnet: Evolution channels gradient descent in super neural networks.
- Galanter, P. (XXXX). Generative art theory. 1:631.
- Gao, J., Li, Z., Nevatia, R., et al. (XXXX). Knowledge concentration: Learning 100k object classifiers in a single cnn.
- Gatys, L. A., Ecker, A. S., and Bethge, M. (XXXX). A neural algorithm of artistic style.
- Gesmundo, A. and Dean, J. (XXXX). munet: Evolving pretrained deep neural networks into scalable auto-tuning multitask systems.
- Gokaslan, A. and Cohen, V. (XXXX). Openwebtext corpus.

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (XXXXa). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (XXXXb). Generative adversarial networks.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. (XXXXc). Explaining and harnessing adversarial examples.
- Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (XXXX). Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.
- Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (XXXXa). Bootstrap your own latent-a new approach to self-supervised learning. 33:21271–21284.
- 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., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (XXXXb). Bootstrap your own latent: A new approach to self-supervised learning.
- Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. (XXXX). Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer.
- Harnad, S. (XXXX). The symbol grounding problem. 42(1-3):335–346.
- Harris, Z. et al. (XXXX). Distributional hypothesis. 10(23):146–162.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (XXXXa). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.
- He, K., Zhang, X., Ren, S., and Sun, J. (XXXXb). Deep residual learning for image recognition.
- Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. (XXXX). Towards the systematic reporting of the energy and carbon footprints of machine learning. 21(248):1–43.
- Herdade, S., Kappeler, A., Boakye, K., and Soares, J. (XXXX). Image captioning: Transforming objects into words. pages 11135–11145.

- Hill, F. and Korhonen, A. (XXXX). Learning abstract concept embeddings from multi-modal data: Since you probably can't see what i mean. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 255–265.
- Hill, F., Reichart, R., and Korhonen, A. (XXXX). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. 41(4):665–695.
- Hinton, G., Vinyals, O., and Dean, J. (XXXX). Distilling the knowledge in a neural network.
- Hochreiter, S. and Schmidhuber, J. (XXXX). Long short-term memory. 9(8):1735–1780.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. (XXXX). Training compute-optimal large language models.
- Hu, R. and Singh, A. (XXXXa). Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449.
- Hu, R. and Singh, A. (XXXXb). Unit: Multimodal multitask learning with a unified transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1419–1429.
- Huang, L., Wang, W., Chen, J., and Wei, X.-Y. (XXXX). Attention on attention for image captioning. pages 4633–4642.
- Huang, S.-C., Pareek, A., Seyyedi, S., Banerjee, I., and Lungren, M. P. (2020). Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *NPJ digital medicine*, 3(1):1–9.
- Hudson, D. A. and Manning, C. D. (XXXX). Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.
- Ive, J., Madhyastha, P., and Specia, L. (XXXX). Distilling translations with visual awareness.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (XXXX). Adaptive mixtures of local experts. 3(1):79–87.
- Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. (XXXX). Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR.

- Jia, C., Yang, Y., Xia, Y., Chen, Y., Parekh, Z., Pham, H., Le, Q. V., Sung, Y., Li, Z., and Duerig, T. (XXXXa). Scaling up visual and vision-language representation learning with noisy text supervision.
- Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (XXXXb). Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.
- Jordan, M. I. and Jacobs, R. A. (XXXX). Hierarchical mixtures of experts and the em algorithm. 6(2):181–214.
- Joshi, G., Walambe, R., and Kotecha, K. (XXXX). A review on explainability in multimodal deep neural nets. 9:59800–59821.
- Kaiser, L., Gomez, A. N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., and Uszkoreit, J. (XXXX). One model to learn them all.
- Karpathy, A. and Fei-Fei, L. (XXXX). Deep visual-semantic alignments for generating image descriptions.
- Karras, T., Laine, S., and Aila, T. (XXXX). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.
- Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):1–12.
- Kiela, D. and Bottou, L. (XXXX). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP)*, pages 36–45.
- Kiela, D., Conneau, A., Jabri, A., and Nickel, M. (XXXX). Learning visually grounded sentence representations.
- Kingma, D. P. and Welling, M. (XXXX). An introduction to variational autoencoders.
- Kiros, J., Chan, W., and Hinton, G. (XXXX). Illustrative language understanding: Large-scale visual grounding with image search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 922–933.
- Koehn, P. (XXXX). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86.

- Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. (XXXX). Large scale learning of general visual representations for transfer. 2(8).
- Kopper, P., Wiegerebe, S., Bischl, B., Bender, A., and Rügamer, D. (2022). DeepPamm: Deep piecewise exponential additive mixed models for complex hazard structures in survival analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 249–261. Springer.
- Kottur, S., Vedantam, R., Moura, J. M., and Parikh, D. (XXXX). Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4985–4994.
- Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M., and Fei-Fei, L. (XXXX). Visual genome: Connecting language and vision using crowd-sourced dense image annotations.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (XXXX). Imagenet classification with deep convolutional neural networks. 25.
- Kudo, T. and Richardson, J. (XXXX). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics.
- Lazaridou, A., Pham, N. T., and Baroni, M. (XXXX). Combining language and vision with a multimodal skip-gram model.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (XXXX). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics.
- Lewkowycz, A., Andreassen, A., Dohan, D. M., Dyer, E. S., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. (XXXX). Solving quantitative reasoning problems with language models.
- Lin, C.-Y. (XXXX). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (XXXXa). Microsoft coco: Common objects in context.

- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (XXXXb). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (XXXXc). Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (XXXX). Roberta: A robustly optimized bert pretraining approach.
- Lottick, K., Susai, S., Friedler, S. A., and Wilson, J. P. (XXXX). Energy usage reports: Environmental awareness as part of algorithmic accountability.
- Lu, J., Batra, D., Parikh, D., and Lee, S. (XXXXa). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.
- Lu, J., Batra, D., Parikh, D., and Lee, S. (XXXXb). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. 32.
- Lu, Y., Zhu, W., Wang, X. E., Eckstein, M., and Wang, W. Y. (XXXXc). Imagination-augmented natural language understanding.
- Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. (XXXX). Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196.
- Mayer, T. and Cysouw, M. (XXXX). Creating a massively parallel bible corpus. 135(273):40.
- Mccormack, J. and Gambardella, C. C. (XXXX). Growing and evolving 3-d prints. 26(1):88–99.
- MICHAEL BARTHEL, GALEN STOCKING, J. H. and MITCHELL, A. (XXXX). Reddit news users more likely to be male, young and digital in their news preferences.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (XXXX). Efficient estimation of word representations in vector space.
- Mineault, P. (XXXX). Unsupervised models of the brain.
- Mircosoft (XXXX). Evaluate:detection.
- Mordvintsev, A. (XXXX). Inceptionism: Going deeper into neural networks.

- Mustafa, B., Riquelme, C., Puigcerver, J., Jenatton, R., and Houlsby, N. (XXXX). Multimodal contrastive learning with limoe: the language-image mixture of experts.
- Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (XXXX). Glide: Towards photorealistic image generation and editing with text-guided diffusion models.
- OpenAI (XXXX). Dall-e.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (XXXX). 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. Association for Computational Linguistics.
- Parcalabescu, L., Cafagna, M., Muradjan, L., Frank, A., Calixto, I., and Gatt, A. (XXXX). VALSE: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8253–8280. Association for Computational Linguistics.
- Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. (XXXX). Styleclip: Text-driven manipulation of stylegan imagery.
- Pennington, J., Socher, R., and Manning, C. D. (XXXX). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Perez, E., Kiela, D., and Cho, K. (XXXX). True few-shot learning with language models. 34:11054–11070.
- Pezzelle, S., Takmaz, E., and Fernández, R. (XXXX). Word representation learning in multimodal pre-trained transformers: An intrinsic evaluation. 9:1563–1579.
- Pölsterl, S., Sarasua, I., Gutiérrez-Becker, B., and Wachinger, C. (2019). A wide and deep neural network for survival analysis from anatomical shape and tabular clinical data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 453–464. Springer.
- Prabhu, V. U. and Birhane, A. (XXXX). Large image datasets: A pyrrhic win for computer vision?
- Qiao, H., Liu, V., and Chilton, L. (XXXX). Initial images: Using image prompts to improve subject representation in multimodal ai generated art. In *Creativity and Cognition*, pages 15–28.

- R Core Team (XXXX). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (XXXXa). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (XXXXb). Learning transferable visual models from natural language supervision.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (XXXXc). Language models are unsupervised multitask learners.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (XXXXd). Language models are unsupervised multitask learners. 1(8):9.
- Rajpurkar, P., Jia, R., and Liang, P. (XXXXa). Know what you don't know: Unanswerable questions for squad.
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (XXXXb). Squad: 100,000+ questions for machine comprehension of text.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (XXXXa). Hierarchical text-conditional image generation with clip latents.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (XXXXb). Zero-shot text-to-image generation.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (XXXXc). Zero-shot text-to-image generation. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.
- Rebuffi, S.-A., Bilen, H., and Vedaldi, A. (XXXX). Learning multiple visual domains with residual adapters. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (XXXX). Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR.
- Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. (XXXX). A generalist agent.

- Ren, S., He, K., Girshick, R., and Sun, J. (XXXX). Faster r-cnn: Towards real-time object detection with region proposal networks. 28.
- Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (XXXX). Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195.
- Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (XXXX). Beyond accuracy: Behavioral testing of nlp models with checklist.
- Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Sustano Pinto, A., Keysers, D., and Houlsby, N. (XXXX). Scaling vision with sparse mixture of experts. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8583–8595. Curran Associates, Inc.
- Rügamer, D., Kolb, C., and Klein, N. (2020). Semi-structured deep distributional regression: Combining structured additive models and deep learning. *arXiv preprint arXiv:2002.05777*.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (XXXX). Imagenet large scale visual recognition challenge. 115(3):211–252.
- Schuhmann, C. (XXXX). Laion-400-million open dataset.
- Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. (XXXX). LAION-400M: open dataset of clip-filtered 400 million image-text pairs.
- Sennrich, R., Haddow, B., and Birch, A. (XXXXa). Neural machine translation of rare words with subword units.
- Sennrich, R., Haddow, B., and Birch, A. (XXXXb). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.
- Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., and Sun, J. (XXXX). Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439.
- Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (XXXX). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer.
- Shekhar, R., Pezzelle, S., Klimovich, Y., Herbelot, A., Nabi, M., Sangineto, E., and Bernardi, R. (XXXX). Foil it! find one mismatch between image and language caption.

- Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., and Keutzer, K. (XXXX). How much can clip benefit vision-and-language tasks?
- Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. (XXXX). The woman worked as a babysitter: On biases in language generation.
- Shonenkov, A. (XXXX). rudall-e.
- Sikarwar, A. and Kreiman, G. (XXXX). On the efficacy of co-attention transformer layers in visual question answering.
- Silberer, C. and Lapata, M. (XXXXa). Grounded models of semantic representation. In *Tsujii J, Henderson J, Pașca M, editors. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning; 2012 Jul 12–14; Jeju Island, Korea. Stroudsburg: ACL; 2012. p. 1423–33*. ACL (Association for Computational Linguistics).
- Silberer, C. and Lapata, M. (XXXXb). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732.
- Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. (XXXX). Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650.
- Socher, R. and Fei-fei, L. (XXXX). Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- Soderlund, J. and Blair, A. (XXXX). Adversarial image generation using evolution and deep learning. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.
- Srinivasan, K., Raman, K., Chen, J., Bendersky, M., and Najork, M. (XXXX). Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2443–2449.
- Srinivasan, R. and Uchino, K. (XXXX). Biases in generative art: A causal look from the lens of art history. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 41–51.

- Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (XXXX). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.
- Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. (XXXX). How to train your vit? data, augmentation, and regularization in vision transformers.
- Strubell, E., Ganesh, A., and McCallum, A. (XXXXa). Energy and policy considerations for deep learning in nlp.
- Strubell, E., Ganesh, A., and McCallum, A. (XXXXb). Energy and policy considerations for deep learning in nlp.
- Sun, Q., Wang, Y., Xu, C., Zheng, K., Yang, Y., Hu, H., Xu, F., Zhang, J., Geng, X., and Jiang, D. (XXXX). Multimodal dialogue response generation.
- Sutton, R. S. (XXXX). The bitter lesson.
- Tan, H. and Bansal, M. (XXXX). Vokenization: Improving language understanding with contextualized, visual-grounded supervision.
- Tan, M. and Le, Q. V. (XXXX). Efficientnet: Rethinking model scaling for convolutional neural networks.
- Tong, C., Li, J., Lang, C., Kong, F., Niu, J., and Rodrigues, J. J. (2018). An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders. *Journal of parallel and distributed computing*, 117:267–273.
- Uppal, S., Bhagat, S., Hazarika, D., Majumder, N., Poria, S., Zimmermann, R., and Zadeh, A. (XXXX). Multimodal research in vision and language: A review of current and emerging trends. 77:149–171.
- Vale-Silva, L. A. and Rohr, K. (2021). Long-term cancer survival prediction using multimodal deep learning. *Scientific Reports*, 11(1):1–12.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (XXXXa). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (XXXXb). Attention is all you need. 30.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (XXXXc). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S.,

- and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Vedantam, R., Zitnick, C. L., and Parikh, D. (XXXX). Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.
- Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (XXXX). Show and tell: A neural image caption generator. pages 3156–3164.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (XXXXa). Glue: A multi-task benchmark and analysis platform for natural language understanding.
- Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. (XXXXb). OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR.
- Wang, Q., Li, R., Wang, Q., and Chen, S. (XXXXc). Non-fungible token (nft): Overview, evaluation, opportunities and challenges.
- Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. (XXXX). Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678.
- Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, E. (XXXX). Ccnet: Extracting high quality monolingual datasets from web crawl data.
- WZRD (XXXX). Wzrd.
- Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (XXXX). Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (XXXX). Show, attend and tell: Neural image caption generation with visual attention.
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (XXXX). mt5: A massively multilingual pre-trained text-to-text transformer.

- Yang, X., Tang, K., Zhang, H., and Cai, J. (XXXX). Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yann, L. and Ishan, M. (XXXX). Self-supervised learning: The dark matter of intelligence.
- Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S.-C. (XXXXa). I2t: Image parsing to text description. 98(8):1485–1508.
- Yao, T., Pan, Y., Li, Y., and Mei, T. (XXXXb). Exploring visual relationship for image captioning.
- Yao, T., Pan, Y., Li, Y., and Mei, T. (XXXXc). Exploring visual relationship for image captioning.
- Yu, J., Xu, Y., Koh, J., Luong, T., Baid, G., Vasudevan, V., Ku, A., Yang, Y., Ayan, B., Hutchinson, B., Han, W., Parekh, Z., Li, X., Zhang, H., Baldridge, J., and Wu, Y. (XXXX). Scaling autoregressive models for content-rich text-to-image generation.
- Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al. (XXXXa). Florence: A new foundation model for computer vision.
- Yuan, S., Shuai, Z., Jiahong, L., Zhao, X., Hanyu, Z., and Jie, T. (XXXXb). Wudaomm: A large-scale multi-modal dataset for pre-training models.
- Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (XXXXa). From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.
- Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (XXXXb). Swag: A large-scale adversarial dataset for grounded commonsense inference.
- Zeng, A., Wong, A., Welker, S., Choromanski, K., Tombari, F., Purohit, A., Ryoo, M., Sindhwani, V., Lee, J., Vanhoucke, V., et al. (XXXX). Socratic models: Composing zero-shot multimodal reasoning with language.
- Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., and Parikh, D. (XXXXa). Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5014–5022.
- Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., and Langlotz, C. P. (XXXXb). Contrastive learning of medical visual representations from paired images and text.

Zhu, X., Yao, J., and Huang, J. (2016). Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 544–547. IEEE.

Zhuang, C., Yan, S., Nayebi, A., Schrimpf, M., Frank, M. C., DiCarlo, J. J., and Yamins, D. L. (XXXX). Unsupervised neural network models of the ventral visual stream. 118(3):e2014196118.